Analysis of High-Order Element Types for Implicit Large Eddy Simulation

Carlos A. Pereira

A Thesis

In the Department

of

Mechanical, Industrial and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Master of Applied Science (Mechanical Engineering) at

Concordia University

July, 2019

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Carlos A. Pereira
Entitled: Analysis of High-Order Element Types for Implicit Large Eddy Simulation

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Mechanical Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Commitee:

_____ Chair
Dr. Behrooz Yousefzadeh

_____ Examiner
Dr. Marius Paraschivoiu

_____ Examiner
Dr. Tristan Glatard

_____ Supervisor
Dr. Brian Vermeire

Approved by _____
Chair of Department or Graduate Program Director

_____ 2019 _____
Dean of Faculty

Abstract

Analysis of High-Order Element Types for Implicit Large Eddy Simulation

Carlos A. Pereira

The use of high-order schemes continues to increase, with current methods becoming more robust and reliable. The resolution of complex turbulent flows using Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS) can be computed more efficiently with high-order methods such as the Flux Reconstruction (FR) approach. We make use of the implicit form of LES, referred to as ILES, in which the numerical dissipation of the spatial scheme passively filters high-frequency modes and no subgrid scale turbulent model is explicitly implemented. Therefore, given the inherent three-dimensional behaviour of turbulent flows, it is important to understand the spectral characteristics of spatial discretizations in three dimensions. The dispersive and dissipative properties of hexahedra, prismatic and tetrahedral element types are compared using Fourier Analysis. This comparison is performed on a per degree of freedom basis to assess their suitability for ILES in terms of computational cost. We observe dispersion relations that display non-smooth behaviour for tetrahedral and prismatic elements, with the presence of jumps in the solution modes. Semilogarithmic plots of the numerical error are presented. We observe that the amount of numerical dissipation and dispersion added by hexahedral elements is the least, followed by prisms and finally tetrahedra. We validate our analysis comparing results obtained on computational domains with comparable computational cost against DNS data. Hexahedral elements have the best agreement with the reference data, followed by prismatic and finally tetrahedral elements, which is consistent with the spectral analysis.

Acknowledgements

I would like to express my gratitude to my parents, Carlos and Amalia, and my sister, Paola, for their love and support during this time. Their efforts and sacrifice have allowed me to pursue this degree.

I would also like to thank the rest of my family for being a true inspiration and for the love they constantly give me.

I would like to thank Ana, Nicolas, Fabiany and Romulo for their emotional support and encouragement to become a better individual each day.

I would like to thank my lab mates for all the time we have enjoyed together. I am grateful to Hamid for always bringing inspirational and stimulating conversations on Aerodynamics to our free time.

I would like to offer my special thanks to Dr. Amine Ben Haj Ali, who guided me in the development of my programming skills during my internship at Bombardier.

Finally, I would like to acknowledge the guidance I have received from my supervisor, Dr. Brian Vermeire. His knowledge of high-order Computational Fluid Dynamics has been an inspiration and have motivated me to pursue a Ph.D. in this field. I am grateful for the time he dedicated to help me define the direction of this research, and for teaching me tools to succeed in this field.

Contents

List of Figures

List of Tables

Acronyms

CFD	Computational Fluid Dynamics
DG	Discontinuous Galerkin
DOF	Degrees of Freedom
DNS	Direct Numerical Simulation
FLOPS	Floating-Point Operations per Second
FR	Flux Reconstruction
FV	Finite Volume
HORUS	High-ORder Unstructured Solver
ILES	Implicit Large Eddy Simulation
LCP	Lifting Collocation Penalty
LDG	Local Discontinuous Galerkin
LES	Large Eddy Simulation
MMS	Method of Manufactured Solutions
MP2	Mammouth Parallel II
OOA	Order of Accuracy
PSD	Power Spectral Density
RANS	Reynolds-Averaged Navier-Stokes
RK	Runge Kutta
SD	Spectral Difference
SDIRK	Singly-Diagonal Implicit Runge-Kutta
SGS	Subgrid-scale
SV	Spectral Volume
TGV	Taylor-Green Vortex

Nomenclature

English Letters

$\boldsymbol{a}$	Acceleration vector
c_d	Drag coefficient
c_l	Lift coefficient
c_p	Specific heat coefficient at constant pressure
c_v	Specific heat coefficient at constant volume
d	Diameter
D	Derivative matrix
e	Internal energy
E	Total energy
E_k	Kinetic energy
$\boldsymbol{f}$	External body force
F	Flux variable
$\boldsymbol{F}$	Flux vector
$f^*_{e,I}$	Riemann flux at interface I for element Ω'_e
$f_{e,k}$	Nodal flux coefficient at point $x_{e,k}$
g	Correction functions in one-dimensional FR
h	Specific enthalpy
H	Total enthalpy
h_e	Element edge length
i	Imaginary number $\sqrt{-1}$
$\overline{\overline{I}}$	Unit tensor
k	Thermal conductivity
L	Domain length
$\tilde{\ell}_k$	Orthonormal Legendre polynomial of degree k
$\hat{\ell}$	Turbulent integral scale size
m	Mass
M	Semidiscrete matrix
M_a	Mach number
n	Time level
N	Number of elements
N_{dim}	Number of dimensions
N_{DOF}	Number of degrees of freedom

N_{sub}	Number of subdivisions in spectral analysis
p	Pressure
P	Polynomial degree
Pr	Prandtl number
$q_{e,k}$	Nodal solution derivative coefficient at point $x_{e,k}$
q_H	Heat source other than conduction
r	Specific gas constant
R	Residual
$R_{I,n}$	Radau polynommial of degree n at opposite interface of I
Re	Reynolds number
Re_b	Bulk velocity Reynolds number
Re_c	Centerline velocity Reynolds number
Re_τ	Friction velocity Reynolds number
S	Source term in conservation laws
t	Time
T	Temperature
t^*	Characteristic time
u	Solution variable
$\boldsymbol{U}$	Eigenvector of semidiscretization M
$\hat{U}$	Matrix of eigenvectors $\boldsymbol{U}$
$u^*_{e,I}$	Common solution at interface I for element Ω'_e
$u_{e,k}$	Nodal solution coefficient at point $x_{e,k}$
$\tilde{u}_{e,k}$	kth modal solution coefficient in element Ω'_e
v	Velocity
V	Vandermonde matrix
v_i	Velocity component in the i-th direction
$v_{i,rms}$	Root-mean-square velocity fluctuation in the i-th direction
v_τ	Wall-friction velocity
x	Cartesian coordinate in physical space
$\boldsymbol{x}$	Coordinate vector in physical space $x = [x_1, x_2, x_3]$
x_L, x_R	One-dimensional element interfaces
W_f	Work from body forces

Greek Letters

$\boldsymbol{\alpha}$	Advection velocity
β	Wavevector angle
γ	Ratio of specific heat coefficients
$\gamma_{c,e}$	Nondimensionalization of wavenumber by computational cost for element Ω_e
Γ	Mapping function
δ	Characteristic length

δ_{ij}	Kronecker delta
Δt	Time step size
Δx	Grid spacing
ϵ	Kinetic energy dissipation rate
κ	Wavenumber
$\boldsymbol{\kappa}$	Wave vector
$\overline{\kappa}$	Nondimensional wavenumber $\bar{\kappa} = \frac{\kappa h}{\gamma_{c,e}}$
λ	Eigenvalue of semidiscrete operator
μ	Dynamic viscosity
ν	Wave frequency
$\tilde{\nu}$	Scalar diffusion coefficient
ξ	Cartesian coordinate in reference space
$\boldsymbol{\xi}$	Coordinate vector in reference space $\boldsymbol{\xi} = [\xi_1, \xi_2, \xi_3]$
$\overline{\overline{\sigma}}$	Stress tensor
τ	Viscous shear stress
ρ	Density
$\boldsymbol{\zeta}$	Vorticity vector
$\hat{\eta}$	Kolmogorov micro-scale
$\boldsymbol{\eta}_e$	Correction field for element Ω_e
θ	Advection velocity angle
ϕ	Nodal basis function
χ	Lifting coefficient for LCP formulation
ψ	Eigenfunction of semidiscretization M
Ω	Computational domain
Ω_c	Cuboidal element in Ω
ω^δ	Numerical frequency of semidiscrete operator M
Ω_e	Domain partion in physical space
Ω'_e	Reference element
$\partial\Omega$	Boundary of control volume

Operators

$\otimes$	Tensor product
$(\bullet)_x$	Partial derivative of $\bullet$ with respect to x
$\boldsymbol{\nabla}$	Divergence operator $\nabla = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right]$
$\|\bullet\|$	Vector magnitude
$\|\bullet\|^2$	L_2 norm

Subscripts

∞	Free-stream condition
C	Convective component
D	Diffusive component

rms	Root-mean square fluctuation
L	Left
R	Right

Superscripts

δ	Discontinuous condition
C	Continuous condition
$+$	Turbulent channel nondimensionalization using v_τ and δ
T	Transpose

Chapter 1

Introduction

Advancements in high-order methods continue to enable accurate and robust simulations of turbulent flows. Complex phenomena can be more efficiently captured with high accuracy compared to current industry-adopted low-order schemes, defined as second-order or lower, which often fail to accurately model complex turbulent flows due to their relatively high numerical error. Hence, high-order schemes are often more suitable for Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES). They can be applied to a wide range of fields of importance such as Computational Fluid Dynamics (CFD), aeroacoustics and other wave propagation phenomena. Some schemes currently used for scale-resolving simulations of complex turbulent flows are the Discontinuous Galerkin (DG) [1, 2, 3], Spectral Volume (SV) [4] and Spectral Difference (SD) [5] methods. A review of these schemes can be found in the works of Huynh [6] and Ekaterinaris [7].

Following NASA's 2030 vision study [8], along with recent progress made in industrial LES using the Flux Reconstruction approach, it is expected that a new family of CFD solvers well-suited to LES on modern many-core hardware architectures will need to be developed. Hence, the solver HORUS (High-ORder Unstructured Solver) is currently under development at the Concordia Computational Aerodynamics Laboratory for this purpose. The development of software tools such as HORUS is critical in the use of CFD for both academic and industrial research and engineering. Along with the development of these unsteady scale-resolving LES tools comes the additional requirement for precise and efficient verification and validation strategies. Due to the innate chaotic unsteadiness of LES, this introduces additional complexities for verification and validation that are not inherent to current industry-standard Reynolds Averaged Navier-Stokes (RANS) solvers. For example, the repeatability of LES simulations is an as-yet unexplored topic in the high-order CFD community due to chaotic divergence of the resulting instantaneous flow field.

As a unifying framework for high-order schemes, Huynh [9] introduced in 2007 the Flux Recon-

struction spatial discretization (FR), which casts conservation laws in divergence form. FR is able to recover the aforementioned high-order schemes with the implementation of particular correction functions that connect the interfaces between elements. Initially developed in one dimension, the FR approach was later extended to multi-dimensions for mixed element types by Wang and Gao [10], three-dimensional problems by Haga and Wang [11], and tetrahedral elements by Williams and Jameson [12]. Hence, enabling its implementation for the computation of turbulent flows.

In addition, the high-order accuracy of FR allows the scheme to be suitable for LES. In this work, we make use of the implicit form of LES, referred to as ILES, in which no explicit sub-grid scale turbulent model (SGS) is implemented. Instead, the numerical error of the spatial discretization passively adds dissipation that models unresolved scales. Furthermore, ILES has shown good agreement with DNS with a significant reduction in computational cost, as presented by Vermeire [13, 14]. The numerical error of the spatial discretization is characterized by dispersion and dissipation behaviour, and it depends on a series of factors, in particular, the polynomial space of the solution and the choice of element type. Furthermore, complex geometries benefit from prismatic and tetrahedral elements due to their flexible nature, making them an appropriate choice for unstructured meshes. Hence, it is important to understand the characteristic behaviour of each element type in order to assess the suitability of the FR approach for ILES using complex geometries. This topic has been widely explored in one and two dimensions for different high-order schemes. However, in three dimensions, it has not been fully characterized. In the following section, we present a compilation of some of the work that has been done in regards to spectral analysis.

1.1 Previous Work on Spectral Analysis

Several studies have been performed on the spectral characteristics of high-order schemes. In one dimension, Moura [15, 16, 17] described the characteristic shape of the dispersion and dissipation curves, noting periodicity in the wavenumber space proportional to the number of points in the solution for DG. The identification of the dominant eigenvalues has been associated with the most energized modes. Vincent et al. [18] proposed an analysis in modal space, in which the dominant component was computed using eigenvalues from all modes. Asthana [19] quantified the distribution of energy amongst all eigenmodes and determined that multiple modes become energized at higher

wavenumbers. Fully discrete analysis has also been performed by Vermeire [20]. In two dimensions, Van den Abeele [21] displayed the dispersion curves on triangular and quadrilateral elements and found periodic behaviour of the dispersion relations as a function of the wave orientation angles for SV and SD schemes. Castonguay et al. [22] described a spectral analysis formulation for triangular meshes. Hu et al. [23] analyzed the numerical properties of quadrilaterals and triangular meshes with different periodic patterns for DG. Recently, Trojak et al. [24] studied the effects of mesh deformation in the quality of meshes composed of quadrilateral elements [24]. In three dimensions, Van den Abeele [25] performed a stability analysis of tetrahedral meshes for the SV method using the matrix method to detect instabilities for modes with long wavelengths.

The aforementioned authors studied the numerical error on different schemes. However, we observed in these studies that high-order methods are generally characterized for yielding more accurate results as the order of the scheme is increased. We will now explore the general advantages of implementing high-order schemes in numerical simulations in terms of accuracy and computational performance.

1.2 Advantages of High-Order Methods

Low-order spatial discretizations, such as the Finite Volume method (FV), are widely used in industry due to their robustness and simplicity. These methods were originally used due to their utility for shocks. However, due to their intrinsically high numerical error, the representation of complex structures becomes expensive. In addition, they mostly rely on the RANS equations, since the use of more accurate approaches, such as DNS and LES, would require grids with extreme resolution. Otherwise, due to the highly dissipative effects of low order schemes, the solution would be highly contaminated when compared to high-order methods, which are more suitable for LES and DNS. Furthermore, high-order schemes require fewer degrees of freedom (DOF) than low-order methods to achieve the same level of accuracy. This is a result of the convergence rate proportional to h^2 for second-order FV schemes, compared to h^n in high-order methods, where n is considered to be at least $n \geq 3$ [7]. This allows high-order methods to utilize coarser meshes than low-order schemes for a given level of accuracy.

While FV methods work well on structured meshes, they normally rely on large stencils that

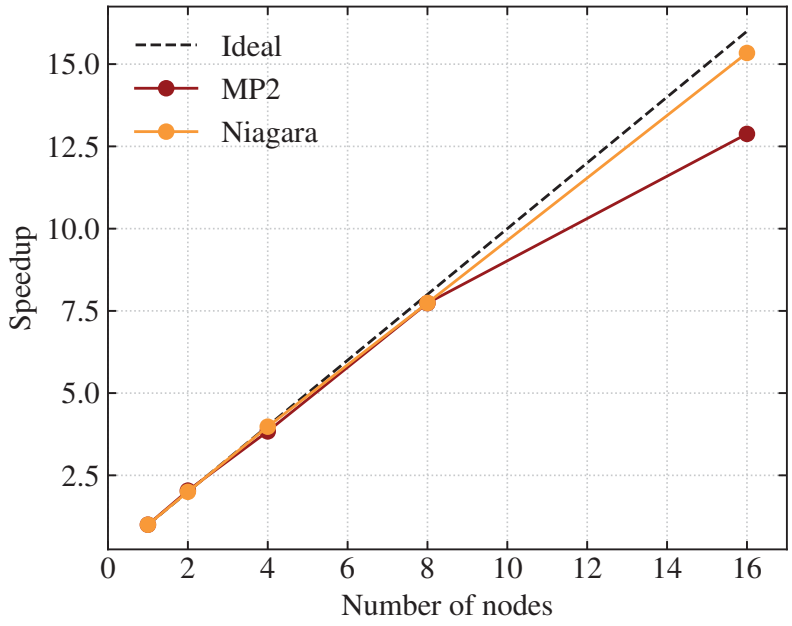


Figure 1.1. Strong scaling study on Compute Canada Clusters Mammouth Parallel II (MP2) and Niagara. Speedup is plotted against number of computer nodes.

limit their performance on unstructured meshes. These methods achieve about 3% of the theoretical floating-point operations per second (FLOPS) on modern architecture due to their high degree of indirect memory access [26]. In contrast, the performance in high-order methods can better leverage today's current computational capabilities. The matrix multiplication nature of these schemes, along with compact stencils that can be obtained with the FR approach, allows for efficient parallel computations using CPUs or GPUs. A quick strong scaling study on Compute Canada's Mammouth Parallel II (MP2) and Niagara clusters, shown in Figure 1.1, exemplifies the speedup gained by increasing the computational resources. We used 1 to 16 nodes, each consisting of 24 and 40 processors for MP2 and Niagara, respectively. The scaling study was performed on a sixth-order spatial FR scheme and the Taylor-Green vortex case, which will be introduced in subsequent sections. The improvement in time-cost metrics is approximately linear, until the communication between parallel processes becomes too expensive given the decreased number of elements assigned to each CPU core. We note that MP2's old interconnect properties are poor compared to the newer cluster Niagara. In this sense, we observe that the performance of FR is dependent on the communication capabilities of the system, and newer hardware will be able to exploit these substantially. Furthermore, the number

of vectorizable (or pointwise direct) operations increases exponentially with the polynomial degree, allowing for more performant schemes when higher orders are used. Hence, schemes such as Flux Reconstruction are able to achieve more than 50% of peak FLOPS [26]. These results have opened doors to exploring turbulent flows with more accuracy by taking advantage of modern hardware. In the following section, we will describe some of the most common turbulence models and their suitability for high-order schemes.

1.3 Turbulence in Numerical Simulations

The chaotic behaviour of turbulent phenomena has led to several mathematical models in an attempt to simulate real-world scenarios, where most flows are turbulent. The use of the RANS equations, DNS and LES are the most common approaches currently used.

RANS methods consist of the time-averaged Navier-Stokes equations. These utilize the concept of Reynolds decomposition to express the instantaneous velocity field as the sum of an averaged and a fluctuation term

$$v(\boldsymbol{x}, t) = \overline{v(\boldsymbol{x})} + v(\boldsymbol{x}, t)'. \tag{1.1}$$

Using this mathematical tool, the momentum equations for incompressible flow can be written as [27]

$$\rho \frac{\overline{D\left(v_j\right)}}{Dt} = \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - \overline{p}\delta_{ij} - \rho\overline{v_i' v_j'}\right], \tag{1.2}$$

where the term $\rho\overline{v_i' v_j'}$ appears from the averaging of the convective term and is referred to as the Reynolds stresses. These terms are typically computed using eddy-viscosity models, which are formulated for specific applications. Due to inherent modelling inaccuracies, the RANS approach often fails for a wide range of applications, including separated and transitional flows. Hence, approaches that capture unsteady flow physics are often needed.

Turbulent flows contain a wide variety of time and length scales. The relationship between the size of the largest eddy and the smallest turbulent scales is a function of the Reynolds number and

is described by the Kolmogorov micro-scale

$$\hat{\eta} = \left(\frac{v\hat{\ell}}{\hat{\mu}}\right)^{\frac{3}{4}} \hat{\ell} = Re^{-\frac{3}{4}}\hat{\ell}, \tag{1.3}$$

where $\hat{\ell}$ and $\hat{\eta}$ are the size of the largest and smallest eddies, respectively and v is the velocity of the smallest scales.

DNS is performed when the computation of every length and time scale within the flow is carried out by integrating the Navier-Stokes equations in time and calculating the instantaneous velocity field at every time step. Nevertheless, the application of DNS is limited by the complexity of geometries and the rapid growth in computational cost. Davidson [28] estimated the number of points that would be required to fully compute all eddies in a three-dimensional box by

$$N_{DOF} = \left(\frac{L}{\Delta x}\right)^3 \approx \left(\frac{L}{\hat{\ell}}\right) Re^{\frac{9}{4}}, \tag{1.4}$$

with L the edge length of a cubic domain of equal sides, and Δx is the average mesh spacing which must be smaller than the Kolmogorov scales for DNS. The number of solution points depends on the Reynolds number. From Equation 1.4 we observe that incrementing Re cubically increases the computational requirements. In addition, the time-step size Δt decreases with the size of the mesh spacing. Therefore, in order to fully resolve flows with medium to large Reynolds number, the amount of computational power required may be unattainable with today's computational capabilities. Hence, current industry requirements may not be able to justify the use of DNS.

In efforts to reduce the computational cost of DNS and still be able to accurately capture pertinent physics, LES was developed. LES is a middle point between RANS modeling and DNS in terms of computational cost. Here, only the largest energy-containing scales are fully computed up until a cut-off wavenumber, which is usually defined by the grid size, as shown in Figure 1.2. Conceptually, the velocity field is decomposed through a filtering operation into a resolved a residual component. As a result, several models have been introduced to account for the under-resolution of the computational model, known as sub-grid scale or SGS models. The Navier-Stokes equations are then manipulated and an SGS stress tensor arises in the momentum equations, which can be explicitly or implicitly modelled. SGS models normally add artificial viscosity, which mimics the

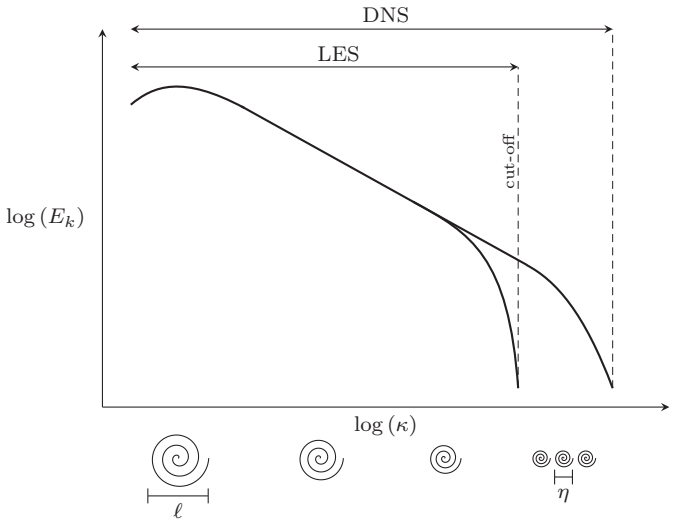


Figure 1.2. Turbulent kinetic energy cascade.

dissipative nature of the smallest turbulent scales.

The resulting LES equations are similar to those obtained by RANS, except with the averaging done in space instead of time, and they capture the interaction between the resolved turbulent scales [29, 30]. In the current research, the implicit form of Large Eddy Simulation is used and no additional SGS model is implemented. Instead, the numerical error of the scheme is allowed mimic the necessary dissipation. This method is called Implicit Large Eddy Simulation (ILES) and has been shown to achieve accurate and consistent results when compared to experiments and DNS [14, 31].

The chaotic behaviour of turbulent flows introduces additional complexities in the application of these models. To ensure results with appropriate numerical accuracy, we must demonstrate their validity and correct implementation by performing verification and validation studies.

1.4 Verification and Validation of CFD software

Verification is a process of great importance in the context of computer simulations. Numerous guides and reports have been made in relation to this topic [32, 33, 34, 35]. Verification is the process of determining whether a mathematical approach resulting from a numerical discretization is able to accurately represent the corresponding mathematical model with sufficient accuracy [36].

With constant advancements in the field of computational fluid dynamics, implementations of new algorithms or enhancements to current codes are carried out on a frequent basis. In

addition, high-order methods particularly benefit from verification procedures to properly justify their implementation and demonstrate their accuracy in terms of computational cost [37]. This is performed by comparing numerical results with existing analytical solutions, such as those for the linear advection, linear diffusion, Euler and Navier-Stokes equations. A simple and convenient procedure is the Method of Manufactured Solutions (MMS), in which the expected solution is initially prescribed and numerically analyzed. In this work, we perform grid convergence studies and compare the numerical results with exact solutions obtained analytically. By looking at error convergence as the grid is refined, we determine whether the order of the spatial or temporal schemes are consistent with their theoretical orders, which verifies the scheme's implementation.

In a similar manner, validation consists in evaluating whether a given mathematical model is able to accurately represent a physical event. To contrast the difference between verification and validation, Roache [38] refers to validation as an interaction between mathematics and physics, whereas verification is purely mathematical and computational-related. In this work, we perform the validation of a large eddy simulation solver using benchmark cases proposed in high-order workshops [39]. This consists of comparing the numerical results with experimental data or highly accurate numerical results, such as those obtained from direct numerical simulation. Some example validation and verification studies can be found in [31, 40, 41].

1.5 Thesis Objectives and Contribution

The objective of this thesis is to conduct a complete analysis of high-order element types and assess their accuracy and suitability in the context of Implicit Large Eddy Simulation. To do this, we will start by performing verification of the in-house High-ORder Unstructured Solver (HORUS) for two and three-dimensional element types. A suite of test verification cases will be proposed to validate components of the solver based on the Navier-Stokes equations. Considering the constant development of the in-house solver, an automatic verification framework will be proposed in the context of a multi-developer environment using version control.

We will then perform spectral analysis of three-dimensional high-order spatial discretizations. We will describe the numerical error introduced by the choice of element type. This will be done by analyzing the numerical dispersion curves of semidiscretizations with hexahedral, prismatic and

tetrahedral elements.

We will validate this spectral analysis using benchmark cases including turbulent channel flow and the Taylor-Green vortex as well as turbulent flow over a cylinder.

Finally, recommendations will be made on the choice of element types for Implicit Large Eddy Simulation.

1.6 Thesis Outline

In Chapter 1, we present the introductory section of this work, including previous work on spectral analysis of spatial discretizations, followed by a general overview of important concepts that form the basis of this thesis.

Chapter 2 contains theoretical concepts in fluid dynamics that are necessary to understand the underlying phenomena in the computational simulations, such as conservation laws.

In Chapter 3, we present the description of the numerical schemes that were the subject of study during this thesis.

Chapter 4 presents verification of the in-house High-ORder Unstructured Solver (HORUS) for both spatial and temporal schemes, as well as accuracy analysis of two and three-dimensional element types.

Chapter 5 is dedicated to spectral analysis of high-order schemes. As an introduction, this analysis is presented in one dimension and later extended to the multidimensional framework.

Chapter 6 presents validation of the spectral analysis in the context of ILES through numerical experiments, including turbulent channel flow, the Taylor-Green vortex and turbulent flow over a cylinder.

Chapter 7 closes this work with some conclusions and recommendations for future work.

An Appendix is provided to include additional dispersion relation for higher-order schemes as well as the computational implementation of algorithms/software developed for this thesis.

Chapter 2

Fundamental Equations

The description of fluid behaviour is defined by a set of conservation laws that together compose the Navier-Stokes equations. In this chapter, we present a brief description of these conservation laws, as well as an introduction to notation and concepts that are relevant to this work, including the linear advection and linear diffusion equations. A more detailed derivation can be found in [42].

2.1 Canonical Form

Consider a region of fluid Ω fixed in space, bounded by a closed surface $\partial\Omega$. The variation of a conserved variable u is associated to the net effects of external sources $S(u,t)$ and how much such a quantity exits or enters through $\partial\Omega$ [42], denoted by the flux vector $\boldsymbol{F}$. The canonical conservation equation can then be written as

$$\frac{\partial}{\partial t}\int_{\Omega} u d\Omega + \int_{\partial\Omega} \boldsymbol{F}(u)\cdot d\boldsymbol{s} = \int_{\Omega} S(u,t) d\Omega, \tag{2.1}$$

where $d\boldsymbol{s}$ is the surface element vector from $\partial\Omega$, pointing in the outward direction. This equation can also be written in divergence form using Gauss theorem, which describes the transformation between surface and volume integrals

$$\int_{\partial\Omega} \boldsymbol{F}(u)\cdot d\boldsymbol{s} = \int_{\Omega} \boldsymbol{\nabla}\cdot\boldsymbol{F}(u) d\Omega. \tag{2.2}$$

Hence, we may rewrite and group similar terms

$$\int_{\Omega}\left(\frac{\partial u}{\partial t} + \boldsymbol{\nabla}\cdot\boldsymbol{F}(u) - S(u,t)\right) d\Omega = 0. \tag{2.3}$$

Since the integrand must vanish, the divergence form of a conservation law is given in canonical form by

$$\frac{\partial u}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{F}(u) = S(u,t). \tag{2.4}$$

2.2 Mass Conservation Equation

Conservation of mass states that mass cannot disappear nor be created. In this sense, the conserved variable is considered to be density, $u = \rho$, and it is linearly proportional to the velocity $\boldsymbol{v}$, which corresponds to the flux without the presence of any source term. The integral form of the conservation law becomes

$$\frac{\partial}{\partial t}\int_{\Omega} \rho d\Omega + \int_{\partial\Omega} \rho(\boldsymbol{v} \cdot d\boldsymbol{s}) = 0. \tag{2.5}$$

By using Gauss theorem, we obtain the divergence form

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v}) = 0. \tag{2.6}$$

2.3 Momentum Conservation Equation

Momentum is a vector quantity defined by the product of the mass m and velocity vector $\boldsymbol{v}$. This equation is a result of the application of Newton's Second Law

$$\boldsymbol{f} = m\boldsymbol{a} = \frac{d}{dt}(m\boldsymbol{v}), \tag{2.7}$$

where m is mass, $\boldsymbol{a}$ is acceleration and $\boldsymbol{f}$ is force. Hence, the conserved variable has magnitude and direction and is now a vector $\boldsymbol{u} = \rho\boldsymbol{v}$. In addition, we must account for the forces that act in the i-th direction due to internal and external forces. This yields

$$\frac{\partial}{\partial t}\int_{\Omega} \rho\boldsymbol{v} d\Omega + \int_{\partial\Omega} \rho\boldsymbol{v}(\boldsymbol{v} \cdot d\boldsymbol{s}) = \int_{\Omega} \rho\boldsymbol{f}_e d\Omega + \int_{\partial\Omega} \overline{\overline{\sigma}} \cdot d\boldsymbol{s}, \tag{2.8}$$

where $\boldsymbol{f}_e$ is a body force and $\overline{\overline{\sigma}}$ is the stress tensor, which can be written as

$$\overline{\overline{\sigma}} = -p\overline{\overline{I}} + \overline{\overline{\tau}}, \tag{2.9}$$

with $\overline{\overline{I}}$ the unit tensor and

$$\tau_{ij} = \left[\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}\right) - \tilde{\lambda}\left(\boldsymbol{\nabla}\cdot\boldsymbol{v}\right)\delta_{ij}\right]. \tag{2.10}$$

Here, δ_{ij} is the Kronecker delta, $\tilde{\lambda}\left(\boldsymbol{\nabla}\cdot\boldsymbol{v}\right)$ is a dilatation term, which is applied to the stresses in normal directions. $\tilde{\lambda} = 2/3$ for Newtonian fluids in equilibrium. After the application of Gauss theorem and some simple rearranging of the right hand side terms, we may cast this equation in divergence form

$$\frac{\partial(\rho\boldsymbol{v})}{\partial t} + \boldsymbol{\nabla}\cdot\left(\rho\boldsymbol{v}\otimes\boldsymbol{v} + p\overline{\overline{I}} - \overline{\overline{\tau}}\right) = \rho\boldsymbol{f}_e. \tag{2.11}$$

2.4 Energy Conservation Equation

The application of the First Law of thermodynamics to a closed system yields the Energy Conservation Law. The total energy E is given by

$$E = e + \frac{1}{2}\left\|\boldsymbol{v}\right\|^2, \tag{2.12}$$

where e is the internal energy, and the second term denotes the kinetic energy. Hence, the conserved variable is $u = \rho E$. Consider a convective flux $\boldsymbol{F}_C = \rho\boldsymbol{v}E$ and a diffusive flux $\boldsymbol{F}_D$ associated with the diffusion of heat in a medium due to molecular conduction. The diffusive flux $\boldsymbol{F}_D$ can be written using Fourier's Law of Heat Conduction, resulting in $\boldsymbol{F}_D = -k\boldsymbol{\nabla}T$, where k is the thermal conduction coefficient and T is the scalar variable temperature. Additionally, we must account for the heat sources other than conduction q_H [42] and the work resulting from the internal forces such as the viscous shear stresses and pressure, denoted by the product of the stress tensor $\overline{\overline{\sigma}}$ and the velocity $\boldsymbol{v}$. This yields

$$\frac{\partial}{\partial t}\int_\Omega \rho E d\Omega + \int_{\partial\Omega}\left(\rho E\boldsymbol{v} - k\boldsymbol{\nabla}T\right)\cdot d\boldsymbol{s} = \int_\Omega\left(\rho\boldsymbol{f}_e\cdot\boldsymbol{v} + q_H\right)d\Omega + \int_{\partial\Omega}\overline{\overline{\sigma}}\cdot\boldsymbol{v}d\boldsymbol{s}, \tag{2.13}$$

where $\overline{\overline{\sigma}}$ is the stress tensor previously defined in Equation 2.9. The divergence form of this equation follows

$$\frac{\partial(\rho E)}{\partial t} + \boldsymbol{\nabla} \cdot \left(\rho \boldsymbol{v} H - k \boldsymbol{\nabla} T - \overline{\overline{\tau}} \cdot \boldsymbol{v}\right) = W_f + q_H, \tag{2.14}$$

where $W_f = \rho \boldsymbol{f}_e \cdot \boldsymbol{v}$ is the work resulting from body forces and H is the stagnation enthalpy given by

$$H = e + \frac{p}{\rho} + \|\boldsymbol{v}\|^2 = h + \left\|\boldsymbol{v}^2\right\| = E + \frac{p}{\rho}, \tag{2.15}$$

in which p is pressure and $\|\cdot\|$ represents a vector magnitude. For an ideal gas

$$p = \rho r T, \tag{2.16}$$

where r is the specific gas constant and T is temperature. The internal energy e and enthalpy h are related to the temperature by

$$e = c_v T, \quad h = c_p T, \tag{2.17}$$

where c_v and c_p are the specific heat coefficients at constant volume and pressure, respectively. These are related by the ratio of specific heat coefficients $\gamma = c_p/c_v$, which is approximately 1.4 for air.

2.5 Euler and Navier-Stokes Equations

In this section, we present the previous conservation laws as part of the set of Euler and Navier Stokes equations. This time, we define fluxes with convective contribution $\boldsymbol{F}_C$ in charge of the transport properties, and fluxes with diffusive contribution $\boldsymbol{F}_D$ due to molecular agitation and associated with viscous effects

$$\frac{\partial \boldsymbol{u}}{\partial t} + \overline{\overline{\boldsymbol{\nabla}}} \cdot (\overline{\overline{\boldsymbol{F}}}_C - \overline{\overline{\boldsymbol{F}}}_D) = \boldsymbol{S}. \tag{2.18}$$

We may also rewrite the divergence form of such set of conservation laws in keeping with vector notation, and commonly used in CFD solver implementations

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial}{\partial x}(\boldsymbol{F}_{C,x} - \boldsymbol{F}_{D,x}) + \frac{\partial}{\partial y}(\boldsymbol{F}_{C,y} - \boldsymbol{F}_{D,y}) + \frac{\partial}{\partial z}(\boldsymbol{F}_{C,z} - \boldsymbol{F}_{D,z}) = \boldsymbol{S}, \tag{2.19}$$

in which the vector of state variables $\boldsymbol{u}$ is given by

$$\boldsymbol{u} = \begin{bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ \rho E \end{bmatrix},$$

where v_i is the i-th component of the velocity vector.

The Euler equations describe the behaviour of compressible fluids in which the effects of viscosity, heat conduction and source terms are negligible. These equations include the previously described conservation laws such as mass, momentum and energy conservation with inviscid and adiabatic assumptions. For this set of equations, we consider only the fluxes with convective effects

$$\boldsymbol{F}_{C,i} = \begin{bmatrix} \rho v_i \\ \rho v_i v_x + \delta_{ix} p \\ \rho v_i v_y + \delta_{iy} p \\ \rho v_i v_z + \delta_{iz} p \\ \rho v_i H \end{bmatrix},$$

where i represents a spatial coordinate $i = [x, y, z]$, and δ_{ij} is the Kronecker delta.

Should the effects of viscosity and heat conduction be considered, we must include the diffusive terms as well as the presence of source terms. This yields the set of Navier-Stokes equations, which describe the behaviour of real-world fluids. The source and diffusive flux terms can be defined by [42]

$$\boldsymbol{S} = \begin{bmatrix} 0 \\ \rho f_x \\ \rho f_y \\ \rho f_z \\ W_f + q_H \end{bmatrix}, \qquad \boldsymbol{F}_{D,i} = \begin{bmatrix} 0 \\ \tau_{xi} \\ \tau_{yi} \\ \tau_{zi} \\ \boldsymbol{\tau}_i \cdot \boldsymbol{v} + k \partial_i T \end{bmatrix},$$

respectively. In these equations, τ is a viscous shear stress, f_i is a body force acting in the i-th

direction and $\partial_i T$ is the derivative of the temperature variable with respect to i, where $i = [x, y, z]$.

2.6 Linear Advection

The linear advection equation describes the passive transport of a scalar quantity u in Ω with advection velocity $\boldsymbol{\alpha}$. This equation is a general form of the mass conservation law.

$$\frac{\partial u}{\partial t} + \boldsymbol{\nabla} \cdot (u\boldsymbol{\alpha}) = 0. \tag{2.20}$$

In this work, we perform spectral analysis by looking at the properties of wave propagation behaviour in semidiscrete systems. The linear advection equation serves as the basis of this study.

2.7 Linear Diffusion

The linear diffusion equation describes a flow's characteristic tendency towards equilibrium. It concerns the distribution of energy in a fluid with velocity $v = 0$ and constant density ρ. In general divergence form, it can be expressed as

$$\frac{\partial u}{\partial t} - \tilde{\nu}\boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} u) = 0, \tag{2.21}$$

where u is a scalar variable such a temperature and $\tilde{\nu}$ is the scalar diffusion coefficient.

Chapter 3

Numerical Schemes

3.1 The Flux Reconstruction Method

3.1.1 Linear Advection

In order to solve the aforementioned systems of equations numerically, a suitable discretization is required. In 2007, Huynh introduced the Flux Reconstruction approach, which is able to recover high-order methods including the DG, SD, and SV schemes by changing a single parameter. With FR, we can discretize equations in divergence form, such as

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0. \tag{3.1}$$

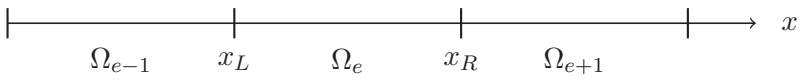


Figure 3.1. One-dimensional domain discretization.

As a first step, we start by diving a computational domain Ω into N_E nonoverlapping elements Ω_e spatially defined by $x \in [x_L, x_R]$, as shown in Figure 3.1. It follows that the formulation is standardized by mapping the physical element Ω_e into a reference space $\xi \in [-1, 1]$, where reference element Ω'_e lives, so the generation of basis functions is simplified. The spatial transformation is done by an invertible point-to-point mapping function, such that

$$x(\xi) = \frac{1}{2}\left(x_L + x_R + \xi h_e\right) \quad \Longleftrightarrow \quad \xi(x) = \frac{2}{h_e}\left(x - \frac{x_L + x_R}{2}\right). \tag{3.2}$$

The metric of this mapping can be then computed

$$\frac{\partial \xi}{\partial x} = \frac{2}{h_e} \quad \Longleftrightarrow \quad \frac{\partial x}{\partial \xi} = \frac{h_e}{2}. \tag{3.3}$$

Within each element Ω_e, the solution is approximated using K nodal values $u^{\delta}_{e,k}$ at locations $x_{e,k}$.

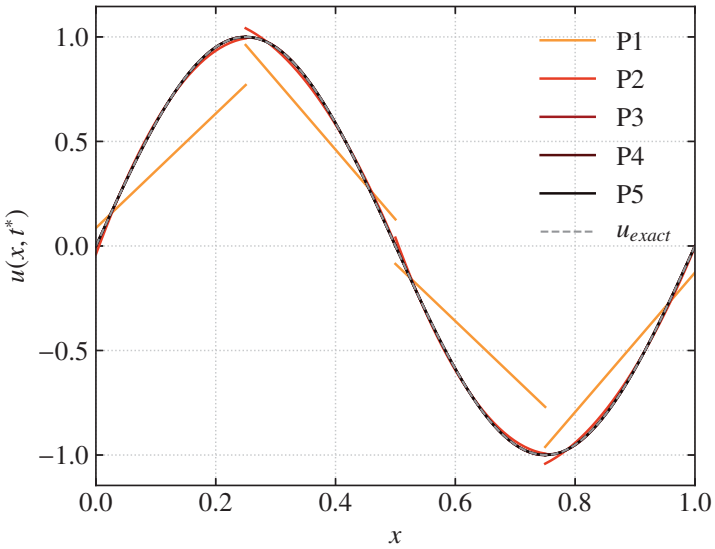


Figure 3.2. Final advection solution $u(x, t = 1)$ for different P-schemes using four elements and $u_0 = \sin(2\pi x)$.

These values may then be used to construct the solution polynomial $u^{\delta}(\xi, t)$ (see Figure 3.2) in polynomial space $P = K - 1$ using Lagrange interpolation

$$u^{\delta}_e(\xi, t) = \sum_{k=1}^{K} u_{e,k}\phi_k(\xi), \tag{3.4}$$

where ϕ_k is the kth nodal basis function, which is given by the Lagrange polynomials in the one-dimensional case. They can be generated by

$$\phi_k(\xi) = \prod_{m=1, i\neq k}^{K} \frac{x - x_m}{x_k - x_m}. \tag{3.5}$$

An interesting feature of these nodal basis functions is that they take the value of 1 at the solution point in question and vanish at all other solution points. An example is shown in Figure 3.3 for different P-schemes using Gauss points. We note the potential of the mapping to reference space, since only one set of nodal basis needs to be generated for the one-dimensional case, assuming all elements interpolate the solution using the same polynomial degree.

The solution polynomial $u^{\delta}_e(\xi, t)$ may also be obtained using a modal approach by means of

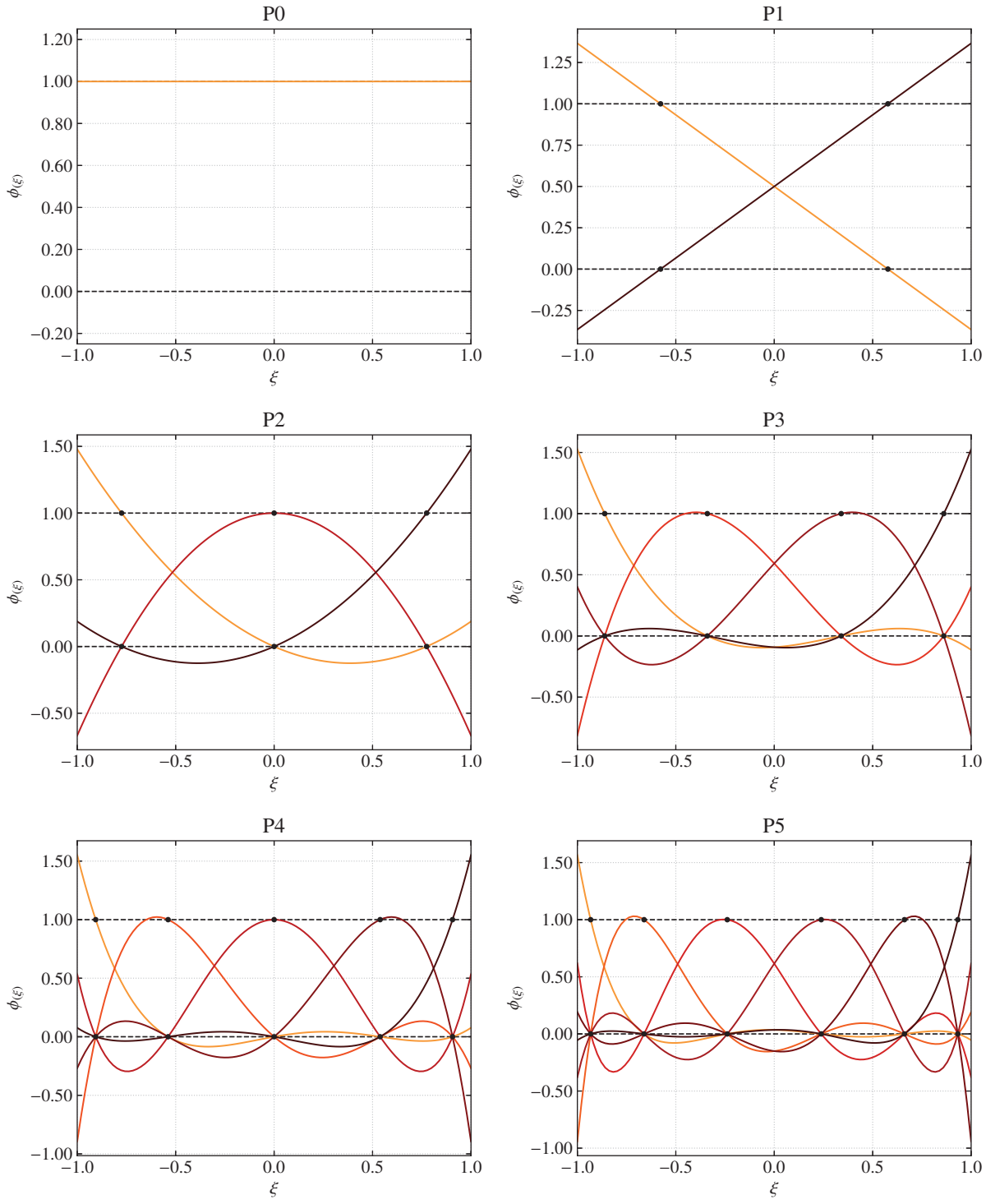


Figure 3.3. Nodal basis functions for the one-dimensional case using Gauss points.

orthonormal basis functions, such as the Legendre polynomials $\tilde{\ell}$ in the one-dimensional case by

$$u_e^{\delta}(\xi, t) = \sum_{k=1}^{K} \tilde{u}_{e,k} \tilde{\ell}_k(\xi), \tag{3.6}$$

where $\tilde{u}_{e,k}$ is the k-th modal expansion coefficient. These polynomials are shown in Figure 3.4. In general, the construction of polynomials is not necessary. We only need the discrete pieces of information which can be condensed in a vector $\boldsymbol{u}_e^{\delta}$ for the modal coefficients and $\tilde{\boldsymbol{u}}_e^{\delta}$ for the nodal coefficients. Hence, the transformation between nodal and modal form can be easily done by means of a Vandermonde matrix V defined as

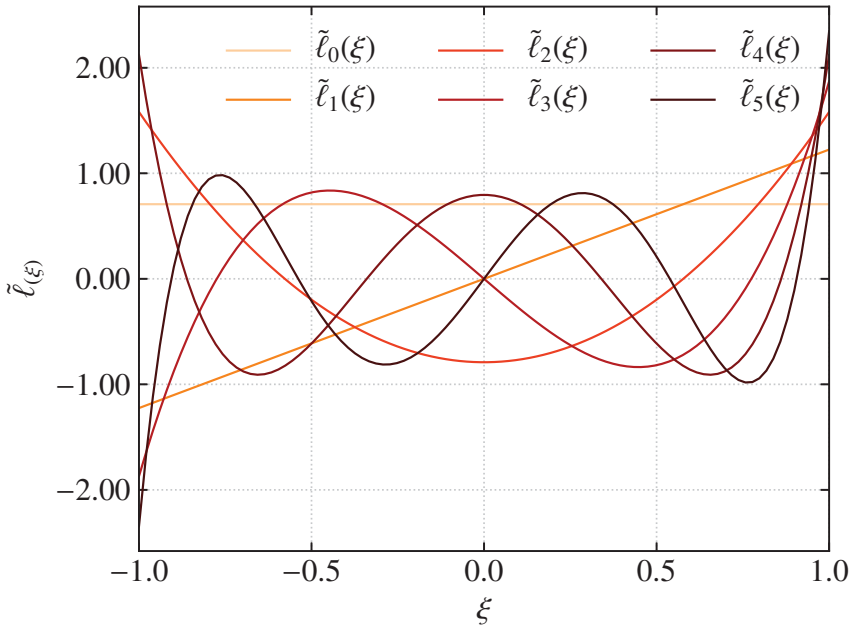


Figure 3.4. Orthonormal Legendre polynomials.

$$V_{ij} = \tilde{\ell}_j(\xi)\Big|_{\xi_i}, \tag{3.7}$$

so that the nodal values can be recovered from the modal coefficients and vice versa, i.e. $\boldsymbol{u}_e^{\delta} = V\tilde{\boldsymbol{u}}_e^{\delta}$. The same way the solution was constructed, a flux polynomial F_e^{δ} inside Ω_e' can be computed using the same nodal bases

$$F_e^{\delta}(\xi, t) = \sum_{k=1}^{K} f_{e,k} \phi_k(\xi), \tag{3.8}$$

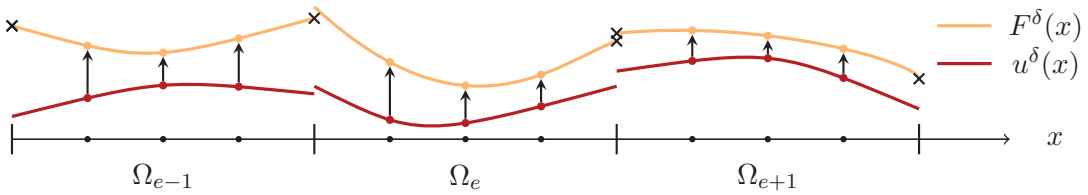


Figure 3.5. Schematic representation of solution and flux polynomials. Circles denote the solution points and crosses the Riemann points.

where $f_{e,k}$ are discrete pieces of data at the same solution points $x_{e,k}$ and can be written compactly in a vector as $\boldsymbol{f}_e^\delta$. Note that F_e^δ is in the same polynomial space of the solution and its derivative would be a degree lower. Since F^δ are discontinuous across cell interfaces, as shown in Figure 3.5, we must reconstruct the flux to obtain a continuous flux polynomial F^C that approximates F^δ and has degree $P+1$. This has a twofold purpose: it enforces conservation in Ω, and ensures the derivative of F^C lies in the same polynomial space of the solution. The construction of F^C is done by the inclusion of a correction flux term η with functions $g(\xi)$ that penalize the existing jumps at the right and left boundaries $x_L,\ x_R$. Hence, for a given element Ω'_e, the continuous flux F_e^C can be expressed as

$$\boldsymbol{F}_e^C = \boldsymbol{f}_e^\delta + \boldsymbol{\eta}_e, \tag{3.9}$$

where n_e is the correction field given by

$$\boldsymbol{\eta}_e = \left[f_{e,L}^* - F_e^\delta(-1)\right] \boldsymbol{g}_L(\xi) + \left[f_{e,R}^* - F_e^\delta(1)\right] \boldsymbol{g}_R(\xi). \tag{3.10}$$

Then, the correction term η_e can be finally calculated using a Riemann solver to compute the common fluxes f_L^* and f_R^*, such as upwind or central schemes[1], and the correction functions $g\,(\xi)$ are evaluated at each node. These must satisfy

$$g_L(-1) = 1, \qquad g_L(1) = 0, \tag{3.11}$$

$$g_R(-1) = 0, \qquad g_R(1) = 1, \tag{3.12}$$

[1]The choice of Riemann solver highly influences the type of numerical error that dominates the numerical solution. An upwind Riemann solver introduces numerical dissipation, whereas a central flux is characteristically dispersive and introduces zero dissipation [19].

and they must be in polynomial space $P+1$. The choice of correction function allows the Flux Reconstruction method to recover existing schemes. For example, the correction functions g_{DG} recover the DG method and are defined by the right Radau polynomials R_R for the left correction function g_L and vice versa. These can be computed in terms of the Legendre polynomials $\tilde{\ell}(\xi)$ [9]

$$g_L = g_{DG,P+1} = R_{R,P+1} = \frac{(-1)^{(P+1)}}{2}\left(\tilde{\ell}_{P+1} - \tilde{\ell}_P\right), \tag{3.13}$$

$$g_R = R_{L,P+1} = g_L(-\xi). \tag{3.14}$$

Next, we need to calculate the derivative of both terms in Equation 3.9. To do so, we first compute the square derivative matrix D of size K, which is standard to all elements

$$D = \begin{bmatrix} \frac{\partial \phi_0(\xi_0)}{\partial \xi} & \cdots & \frac{\partial \phi_k(\xi_0)}{\partial \xi} \\ \vdots & \ddots & \vdots \\ \frac{\partial \phi_0(\xi_k)}{\partial \xi} & \cdots & \frac{\partial \phi_k(\xi_k)}{\partial \xi} \end{bmatrix}.$$

Hence, we obtain the derivative of the discontinuous flux term

$$\left(\boldsymbol{f}_e^{\delta}\right)_{\xi} = D\boldsymbol{f}_e^{\delta}, \tag{3.15}$$

In order to obtain the divergence of the correction field $\boldsymbol{\eta}_e$, we simply need to obtain the derivative of the correction functions, i.e. $(g)_{\xi}$, and evaluate them at each solution point. This can be written compactly

$$\boldsymbol{g}_{\xi} = \begin{bmatrix} (g)_{\xi}(\xi_0) \\ \vdots \\ (g)_{\xi}(\xi_k) \end{bmatrix}.$$

Hence,

$$(\boldsymbol{\eta}_e)_{\xi} = \left[f_{e,L}^{*} - F_e^{\delta}(-1)\right]\boldsymbol{g}_{\xi,L} + \left[f_{e,R}^{*} - F_e^{\delta}(1)\right]\boldsymbol{g}_{\xi,R}, \tag{3.16}$$

where $(\cdot)_{\xi}$ indicates the derivative with respect to ξ. Finally, the global results can be obtained by the use of the metrics in Equation 3.3

$$\left(\boldsymbol{F}_e^C\right)_x = \frac{2}{h_e}\left[\left(\boldsymbol{f}_e^\delta\right)_\xi + \left(\boldsymbol{\eta}_e\right)_\xi\right]. \tag{3.17}$$

3.1.2 Linear Diffusion

Consider the linear diffusion equation

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0, \qquad F = F\left(u, (u)_x\right), \tag{3.18}$$

where $u = u(x,t)$ is the scalar solution variable with derivative $(u)_x$, F is the flux $F = \tilde{\nu}\,(u)_{xx}$ with $\tilde{\nu}$ the scalar diffusion coefficient, x is the spatial coordinate and t is time.

Following Huynh [43], we divide the computational domain Ω into a set of nonoverlapping elements Ω_e with physical coordinate $x \in [x_L, x_R]$. These elements are mapped to a reference space with coordinate $\xi \in [-1,1]$ with element length h_e for efficiency reasons, similar to the discretization of first-order derivatives. The procedure for a second-order derivative requires that we apply a similar approach to the previous section with an intermediate step. In this sense, we rewrite the equation, such that

$$\frac{\partial u}{\partial t} + \frac{\partial q}{\partial x} = 0, \tag{3.19}$$

with $q = \frac{\partial u}{\partial x}$.

Again, the K solution values interpolate a polynomial of degree $P = K-1$ by means of Lagrange interpolation

$$u_e^\delta(\xi, t) = \sum_{k=1}^{K} u_{e,k}\phi_k(\xi). \tag{3.20}$$

As a first step, we discretize q. Contrary to the linear advection discretization, the solution across element cells is expected to be continuous so that its derivative contains data from the neighbouring elements. This is done by following the reconstruction procedure described in the previous section, in which a correction field is applied to account for the cell jumps as well as increase the degree of the solution to $P+1$ so its derivative lies in expected polynomial space P. Hence, the continuous solution polynomial $u_e^C(\xi)$ can be computed from

$$u_e^C(\xi) = u_e^\delta(\xi) + \left[u_L^* - u^\delta(-1)\right] g_L(\xi) + \left[u_R^* - u^\delta(1)\right] g_R(\xi), \tag{3.21}$$

where $u_L^*,\ u_R^*$ are the common solution values across cell interfaces. The first derivative of $u_e^C(\xi)$, $\left(u_e^C\right)_\xi(\xi)$, is then given by

$$\left(u_e^C\right)_\xi(\xi) = \left(u_e^\delta\right)_\xi(\xi) + \left[u_L^* - u^\delta(-1)\right](g_L)_\xi(\xi) + \left[u_R^* - u^\delta(1)\right](g_R)_\xi(\xi), \tag{3.22}$$

where the correction functions are defined similarly to the advection discretization, i.e. using the Radau polynomials recovers a DG scheme.

The common values must be defined by considering an appropriate Riemann solver. Some common choices for the diffusion equation are Cockburn and Shu's [44] one-sided Local Discontinuous Galerkin (LDG) approach, in which at interface I

$$u_I^* = u_L \quad \text{or} \quad u_I^* = u_R, \tag{3.23}$$

and Bassi and Rebay's BR1 and BR2 schemes [45, 46]. However, the BR1 scheme is known for achieving FR's expected orders of accuracy of $P+1$ only for odd P. Therefore, we only consider the BR2 scheme, which follows a central approach such that for a given interface I

$$u_I^* = \frac{1}{2}(u_L + u_R). \tag{3.24}$$

Hence, at each solution point, the corrected derivative [43] can be computed by mapping Equation 3.22 to physical space

$$q_{e,k} = (u)_x(x_{e,k}) = \frac{2}{h_e}\left(u_e^C\right)_\xi(\xi_k). \tag{3.25}$$

These values interpolate a polynomial $q_e^\delta(\xi)$ of degree $K-1$ and is generally discontinuous across cells. Next, we compute the second derivative of the solution $\left(q_e^C\right)_\xi(\xi)$ following a similar procedure

$$\left(q_e^C\right)_\xi(\xi) = \left(q_e^\delta\right)_\xi(\xi) + \left[q_L^* - q^\delta(-1)\right](g_L)_\xi(\xi) + \left[q_R^* - q^\delta(1)\right](g_R)_\xi(\xi). \tag{3.26}$$

In this case, we must define a common derivative q_I^* at each interface I. If the one-sided LDG approach was chosen to compute the common solution values u_L^*, u_R^*, we must select the opposite interface from that chosen for u_I^* to have an equation that is consistent with diffusion behaviour [43]. However, the resulting stencil includes four cells, and it is not convenient in terms of computational

implementation, since FR schemes are generally compact. In this work, we make use of the BR2 scheme as shown in Equation 3.24. Therefore, the computation of the common derivative depends only on information from a given interface at a time, which yields a compact stencil with a high degree of locality.

Finally, applying the reconstruction procedure, we obtain the corrected second derivative by

$$(u)_{xx}(x_{e,k}) = \left(q_e^C\right)_x (x_{e,k}) = \frac{2}{h_e}\left(q_e^C\right)_\xi (\xi_k). \tag{3.27}$$

3.1.3 Multidimensional Formulation

Following the formulation on one-dimensional grids, we present the extension to multi-dimensions. Hence, consider the multidimensional linear advection equation.

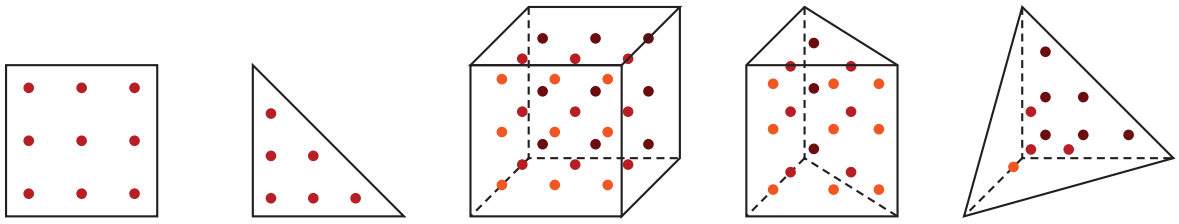
$$\frac{\partial u}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{F} = 0, \tag{3.28}$$

where $u = u(\boldsymbol{x}, t)$ is the solution, $\boldsymbol{F} = \boldsymbol{F}(u)$ is the flux and t is time.

The FR approach starts by dividing the computational domain into N_E elements Ω_e. For the sake of simplicity, these elements are transformed into a reference space element Ω_e' with coordinates $\boldsymbol{\xi}$, where $\boldsymbol{\xi} = [\xi_1, \xi_2, \xi_3]$. The transformation of these elements into the reference space is done by an invertible mapping function, such that

$$\boldsymbol{x} = \Gamma_e(\boldsymbol{\xi}) \quad \Longleftrightarrow \quad \boldsymbol{\xi} = \Gamma_e^{-1}(\boldsymbol{x}), \tag{3.29}$$

where Γ_e is the mapping function.



(a) Quadrilateral (b) Triangle (c) Hexahedron (d) Prism (e) Tetrahedron

Figure 3.6. Schematic view of solution points for $P2$ schemes on three-dimensional element types.

We then define the number of degrees of freedom (N_{DOF}) that support an appropriate polynomial

Table 3.1. Number of degrees of freedom per element type.

Element type	N_{DOF}
Hexahedron	$(P+1)^3$
Prism	$(P+1)^2(P+2)/2$
Tetrahedron	$(P+1)(P+2)(P+3)/6$

of degree P within the reference element Ω_e' (see Table 3.1). The solution is now approximated by a set of piecewise polynomials using nodal basis functions constructed at the N_{DOF} solution points, and is allowed to be discontinuous across element interfaces. From a linear combination of nodal basis functions $\phi_k(\boldsymbol{\xi})$ and nodal coefficients $u_{e,k}$, we can recover the solution polynomial in the local element $u_e^\delta(\boldsymbol{\xi}, t)$

$$u_e^\delta(\boldsymbol{\xi}, t) = \sum_{k=1}^{N_{DOF}} u_{e,k}\phi_k(\boldsymbol{\xi}). \tag{3.30}$$

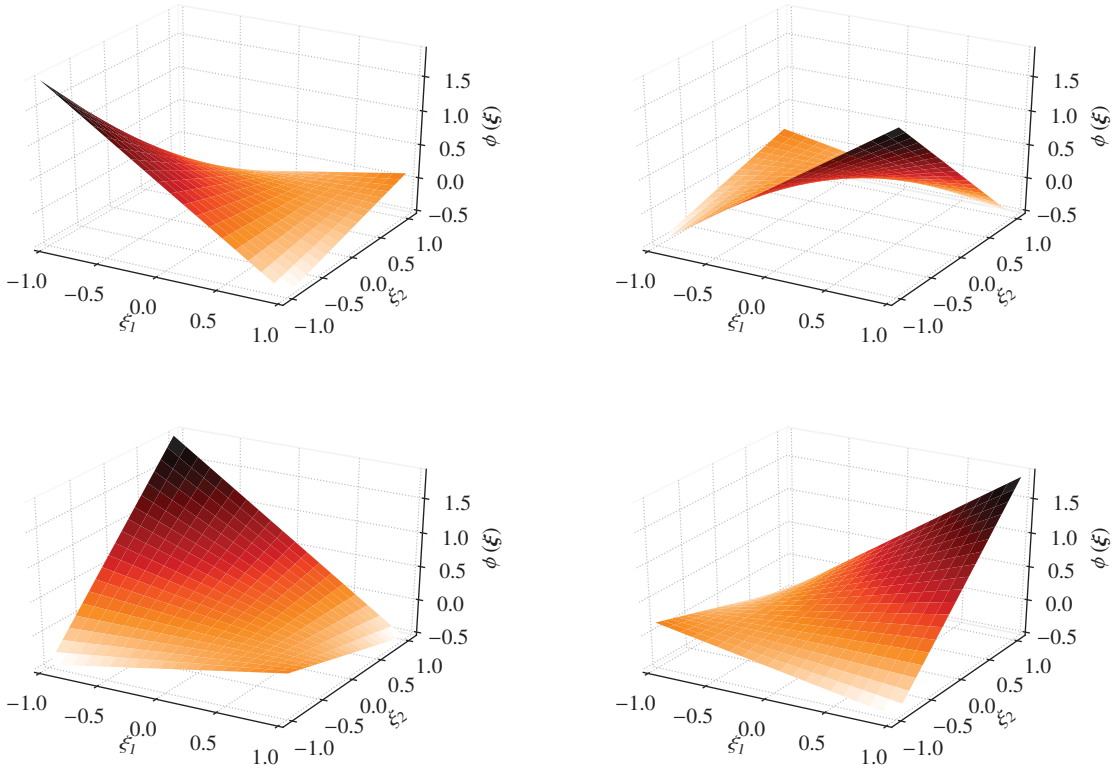


Figure 3.7. Nodal basis functions for quadrilateral elements on $P1$ schemes using tensor product of Gauss points.

A polynomial approximation of the flux may also be interpolated with the same set of nodal basis functions and the nodal value of the flux $\boldsymbol{f}_{e,k}$. Hence,

$$\boldsymbol{F}_e^{\delta}(\boldsymbol{\xi}, t) = \sum_{k=1}^{N_{DOF}} \boldsymbol{f}_{e,k}\phi_k(\boldsymbol{\xi}), \tag{3.31}$$

where $F_e^{\delta}(\boldsymbol{\xi}, t)$ is the discontinuous flux polynomial.

Now, we reconstruct the elementwise continuous flux by a function $F_e^{C}(\boldsymbol{\xi}, t)$, which approximates the discontinuous flux polynomial F_e^{δ}. To account for the jumps across cells, we follow the Lifting Collocation Penalty (LCP) formulation by Wang and Gao [10] for simplex elements. Since the conservation law must also be satisfied at each solution point k, we may rewrite

$$\frac{du_{e,k}^{\delta}}{dt} + \boldsymbol{\nabla} \cdot \boldsymbol{F}_e^{\delta}\left(u_e^{\delta}\right)\Big|_k + \eta_{e,k} = 0, \tag{3.32}$$

where $\nabla \cdot \boldsymbol{F}_e\left(u_e^{\delta}\right)\Big|_k$ denotes the flux divergence evaluated at the solution point k and $\eta_{e,k}$ is a scalar correction field at point k in Ω_e that ensures a globally continuous flux polynomial. The correction field can be determined by

$$\eta_{e,k} = \frac{1}{|\Omega_e|} \sum_{f \in \partial\Omega_e} \sum_{n} \chi_{k,f,n} \left[\tilde{F}\right]_{f,n} S_f, \tag{3.33}$$

where $\chi_{k,f,n}$ is a constant lifting coefficient independent of the solution and geometry that penalizes the normal flux jump $\left[\tilde{F}\right]_{f,n}$ at the cell interfaces at flux point n, S_f is the area of face f and $|\Omega_e|$ is the volume of element e. In this study, consider $\left[\tilde{F}\right]_{f,n}$ calculated with an upwind Riemann solver and the constant lifting coefficients $\chi_{k,f,n}$ that recover the DG scheme [10]. The solution at each DOF may now be updated using a temporal discretization, such as a Runge-Kutta method, for the temporal derivative in Equation 3.32.

3.2 Time-Marching Schemes

RK methods are a family of multi-stage implicit and explicit schemes. We can temporally discretize an equation of the form

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{R} = 0, \tag{3.34}$$

where $u = u(\boldsymbol{x}, t)$ is the solution vector, $\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{u}(\boldsymbol{x}, t))$ is the residual and t is time. At time level n, we start by approximating the solution at s intermediate stages $u_i,\ i = 1, 2 \ldots, s$. These are the result of a linear combination of residuals R_i weighted at all stages, and the value of the solution at time level n using constant values a_{ij}, which are defined by the RK scheme's stage

$$\boldsymbol{u}_i^n = \boldsymbol{u}^n + \Delta t \sum_{j=1}^{s} a_{ij} \boldsymbol{R}_j \quad i = 1, 2, \ldots, s. \tag{3.35}$$

Then, we evaluate the solution at the next time level $n + 1$ with a linear combination of constants b_i

$$\boldsymbol{u}_i^{n+1} = \boldsymbol{u}^n + \Delta t \sum_{i=1}^{s} b_i \boldsymbol{R}_i \quad i = 1, 2, \ldots, s. \tag{3.36}$$

$\boldsymbol{c}$	A
	$\boldsymbol{b}$

Table 3.2. Butcher tableau representation.

These methods can be written compactly using Butcher tableau notation, as shown in Table 3.2, where A is the matrix that contains the coefficients a_{ij}, $\boldsymbol{b}$ is the vector that contains the coefficients b_i and c is a vector with entries for the relative location of the stages at each intermediate stage.

Explicit methods contain lower-diagonal matrices A since the solution at the next time level is predicted from the solution at previous time levels. These methods are memory-efficient and tend to be fast, however, they are constrained by the time step size and are not recommended for very stiff problems.

Schemes with diagonal entries in matrix A are called diagonally-implicit, in which the solution at the next time level also requires information from that time level, resulting in a system of equations. These schemes are known for their stability benefits since large time-step sizes can be taken while maintaining robustness. However, they may be slow due to the required solution of linear systems. In addition, their memory consumption can become significantly elevated for grids with medium to high numbers of degrees of freedom.

Some example Butcher tableaus for the temporal schemes considered in this work are shown

hereafter, specifically the explicit fourth-order Runge-Kutta method with four stages [47] in Table 3.3, and two Singly-Diagonal Implicit Runge-Kutta (SDIRK) [48] schemes with second and third-order accuracy in Tables 3.4 and 3.5, respectively [49].

Table 3.3. Butcher tableau of RK_{44}.

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Table 3.4. Butcher tableau of $SDIRK_{22}$.

0	0.2928932188	0
1	0.7071067811	0.2928932188
	0.7071067811	0.2928932188

Table 3.5. Butcher tableau of $SDIRK_{33}$.

0	0.4358665215	0	0
0.7179332607	0.2820667392	0.4358665215	0
1	1.2084966490	−0.644363171	0.4358665215
	1.2084966490	−0.644363171	0.4358665215

Chapter 4

Verification

Verification is a crucial step in ensuring the correct implementation of numerical schemes. In this chapter, we present convergence studies to verify the advection and diffusion components of the FR approach as well as the Navier-Stokes equations through simple analytical solutions such as planar Couette flow. In addition, we present verification of temporal schemes for the sake of completeness. We perform the study by evaluating the orders or accuracy of different P-schemes using different two and three-dimensional element types, noting that Flux Reconstruction schemes are typically observed to achieve an order of accuracy of $P + 1$, where P is the degree of the solution polynomial.

4.1 Spatial

4.1.1 Linear Advection

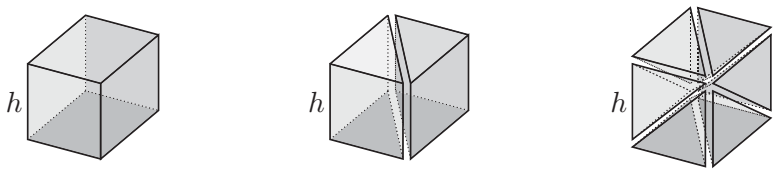
We perform verification using the three-dimensional linear advection equation, which is the basis of the spectral analysis presented in Chapter 5, in which the propagation properties of wave-like solutions are analyzed. Since the exact solution can be determined, we can easily assess its correct implementation by comparing the exact solution with the numerical results.

The linear advection equation describes the translation of a scalar field $u(\boldsymbol{x}, t)$ with constant velocity vector $\boldsymbol{\alpha}$. A periodic hexahedral computational domain of equal edge length L in all directions was set up to verify the equation for spatial discretizations of second to sixth order using the fourth-order Runge-Kutta $RK_{4,4}$ method to minimize temporal error. The flow was initialized with an exponential function

$$u(\boldsymbol{x}, 0) = \prod_{i=1}^{N_{dim}} e^{-\frac{L}{10}(x_i - x_{i_c})^2}, \tag{4.1}$$

where x_{i_c} is simply $L/2$, x_i is the variable coordinate in the ith direction. We present verification for

$N_{dim} = 2$ and $N_{dim} = 3$. For the two-dimensional case, we consider a square domain composed of quadrilateral elements and their subdivisions into triangles. In three dimensions, the computational domain was discretized using regular meshes composed of N_c rectangular cuboids in each direction. Subsequently, these three-dimensional referential elements were subdivided into N_{sub} elements according to Figure 4.1. The simulations were run for one complete cycle $t^* = \frac{L}{\|\boldsymbol{\alpha}\|}$ with advection speed components equal to unity. Figure 4.3 shows the convergence of the L_2 error versus the number of elements, calculated using the discrete norm from Equation 4.2. See Tables 4.1 and 4.2 for details. Note that the error was computed using an 11^{th}-order Gaussian quadrature.



Hexahedron Prisms Tetrahedra

Figure 4.1. Subdivision of rectangular cuboids for three-dimensional cases.

$$\|u\|^{2,\delta} = \sum_{e=1}^{N_E} \|u\|^2_{\Omega_e} . \tag{4.2}$$

As the mesh is refined, the solution converges to the right orders of accuracy, verifying the FR approach for multidimensional advection. In addition, Figure 4.4 shows the L_2 norm of the error against the number of degrees of freedom per direction. By looking at these plots, we observe the benefits of increasing the polynomial degree: high-order schemes require fewer solution points to achieve the same level of error compared to lower-order polynomials. Furthermore, we compare the accuracy of the element types for advection-dominated problems.

In two dimensions, quadrilateral elements show higher accuracy compared to triangles. In three dimensions, hexahedral and prismatic elements are more accurate than tetrahedra on a per-degree of freedom basis. This becomes more prominent for the highest orders, where the error is smaller for the hexahedra by a factor of 100 compared to tetrahedral elements for the finest grid levels considered.

4.1.2 Linear Diffusion

The verification of linear diffusion is performed in this section using MMS. This procedure will provide further support for the numerical experiments realized in Chapter 6. Consider the

Table 4.1. L_2 norm of the solution error for two-dimensional linear advection.

		Quadrilaterals		Triangles	
P	N_c	L_2 Error	Order	L_2 Error	Order
1	10	$\mathbf{2.18} \times 10^{-1}$	-	$\mathbf{1.89} \times 10^{-1}$	-
	20	$\mathbf{6.68} \times 10^{-2}$	1.70	$\mathbf{6.53} \times 10^{-2}$	1.53
	40	$\mathbf{1.23} \times 10^{-2}$	2.44	$\mathbf{1.32} \times 10^{-2}$	2.31
	80	$\mathbf{1.89} \times 10^{-3}$	2.70	$\mathbf{2.01} \times 10^{-3}$	2.72
2	10	$\mathbf{3.07} \times 10^{-2}$	-	$\mathbf{3.92} \times 10^{-2}$	-
	20	$\mathbf{2.24} \times 10^{-3}$	3.78	$\mathbf{3.62} \times 10^{-3}$	3.44
	40	$\mathbf{1.97} \times 10^{-4}$	3.50	$\mathbf{4.05} \times 10^{-4}$	3.16
	80	$\mathbf{1.81} \times 10^{-5}$	3.45	$\mathbf{6.10} \times 10^{-5}$	2.73
3	10	$\mathbf{2.24} \times 10^{-3}$	-	$\mathbf{4.97} \times 10^{-3}$	-
	20	$\mathbf{9.86} \times 10^{-5}$	4.51	$\mathbf{2.07} \times 10^{-4}$	4.58
	40	$\mathbf{6.07} \times 10^{-6}$	4.02	$\mathbf{1.29} \times 10^{-5}$	4.00
	80	$\mathbf{3.81} \times 10^{-7}$	4.00	$\mathbf{3.81} \times 10^{-7}$	5.09
4	10	$\mathbf{3.00} \times 10^{-4}$	-	$\mathbf{5.51} \times 10^{-4}$	-
	20	$\mathbf{7.03} \times 10^{-6}$	5.42	$\mathbf{1.41} \times 10^{-5}$	5.29
	40	$\mathbf{2.21} \times 10^{-7}$	4.99	$\mathbf{4.94} \times 10^{-7}$	4.84
	80	$\mathbf{6.95} \times 10^{-9}$	4.99	$\mathbf{1.72} \times 10^{-8}$	4.84
5	10	$\mathbf{9.42} \times 10^{-6}$	-	$\mathbf{6.65} \times 10^{-5}$	-
	20	$\mathbf{4.19} \times 10^{-7}$	4.49	$\mathbf{1.04} \times 10^{-6}$	6.00
	40	$\mathbf{6.56} \times 10^{-9}$	6.00	$\mathbf{1.65} \times 10^{-8}$	5.98
	80	$\mathbf{1.03} \times 10^{-10}$	5.99	$\mathbf{6.54} \times 10^{-10}$	4.66

Table 4.2. L_2 norm of the solution error for three-dimensional linear advection.

		Hexahedra		Prisms		Tetrahedra	
P	N_c	L_2 Error	Order	L_2 Error	Order	L_2 Error	Order
1	5	4.44×10^{-1}	—	3.34×10^{-1}	—	4.11×10^{-1}	—
	10	2.36×10^{-1}	**0.91**	1.83×10^{-1}	**0.87**	2.28×10^{-1}	**0.85**
	20	7.30×10^{-2}	**1.69**	6.64×10^{-2}	**1.46**	7.43×10^{-2}	**1.62**
	40	1.31×10^{-2}	**2.48**	1.39×10^{-2}	**2.26**	1.42×10^{-2}	**2.38**
2	5	1.79×10^{-1}	—	1.53×10^{-1}	—	1.72×10^{-1}	—
	10	3.07×10^{-2}	**2.54**	4.36×10^{-2}	**1.81**	3.41×10^{-2}	**2.34**
	20	2.06×10^{-3}	**3.90**	4.50×10^{-3}	**3.28**	2.45×10^{-3}	**3.80**
	40	1.55×10^{-4}	**3.73**	5.57×10^{-4}	**3.01**	1.95×10^{-4}	**3.65**
3	5	6.79×10^{-2}	—	7.49×10^{-2}	—	7.32×10^{-2}	
	10	2.28×10^{-3}	**4.90**	7.07×10^{-3}	**3.41**	3.85×10^{-3}	**4.25**
	20	7.60×10^{-5}	**4.90**	3.61×10^{-4}	**4.29**	1.45×10^{-4}	**4.73**
	40	4.52×10^{-6}	**4.07**	2.42×10^{-5}	**3.90**	8.37×10^{-6}	**4.12**
4	5	1.14×10^{-2}	—	3.01×10^{-2}	—	1.43×10^{-2}	—
	10	2.25×10^{-4}	**5.65**	9.50×10^{-4}	**4.98**	8.28×10^{-4}	**4.10**
	20	5.21×10^{-6}	**5.44**	2.62×10^{-5}	**5.18**	2.03×10^{-5}	**5.35**
	40	1.62×10^{-7}	**5.00**	9.74×10^{-7}	**4.75**	5.14×10^{-7}	**5.30**
5	5	2.16×10^{-3}	—	1.10×10^{-2}	—	4.75×10^{-3}	—
	10	3.78×10^{-5}	**5.83**	1.23×10^{-4}	**6.48**	9.78×10^{-5}	**5.60**
	20	3.52×10^{-7}	**6.75**	2.04×10^{-6}	**5.91**	3.17×10^{-6}	**4.95**
	40	4.77×10^{-9}	**6.21**	3.47×10^{-8}	**5.87**	3.91×10^{-8}	**6.34**

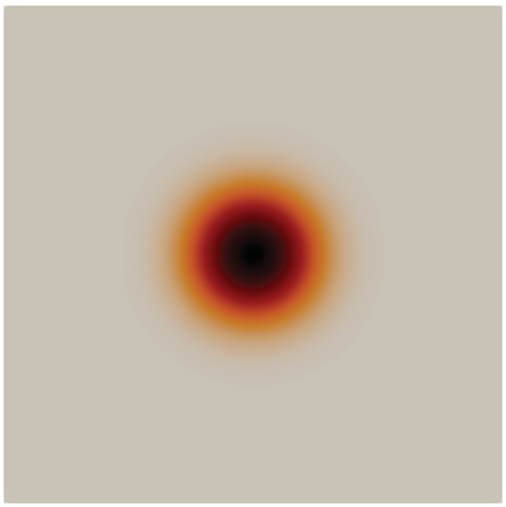


Figure 4.2. Final linear advection solution using $P4$ quadrilateral elements.

steady-state linear diffusion equation with a source term $S(\boldsymbol{x})$

$$-\tilde{\nu} \cdot \nabla^2 u\left(\boldsymbol{x}\right) = S(\boldsymbol{x}), \tag{4.3}$$

where $\tilde{\nu}$ is a scalar diffusion coefficient. A computational domain of equal edge length $L = 2\pi$ is considered. We prescribe a manufactured solution

$$u\left(\boldsymbol{x}\right) = \prod_{i=1}^{N_{dim}} \sin\left(x_i\right), \tag{4.4}$$

with the corresponding source term

$$S(\boldsymbol{x}) = \tilde{\nu} N_{dim} \prod_{i=1}^{N_{dim}} \sin\left(x_i\right). \tag{4.5}$$

The simulation was initialized with $u_0(\boldsymbol{x}) = 0$ and the solver was allowed to diffuse the solution to steady-state conditions using a BR2 scheme [45] for the common fluxes and implicit Euler for the temporal discretization. Table 4.4 and Figure 4.5 display the convergence of the L_2 norm for all element types using Equation 4.2. The convergence in Tables 4.3 and 4.4 concurs with the expected

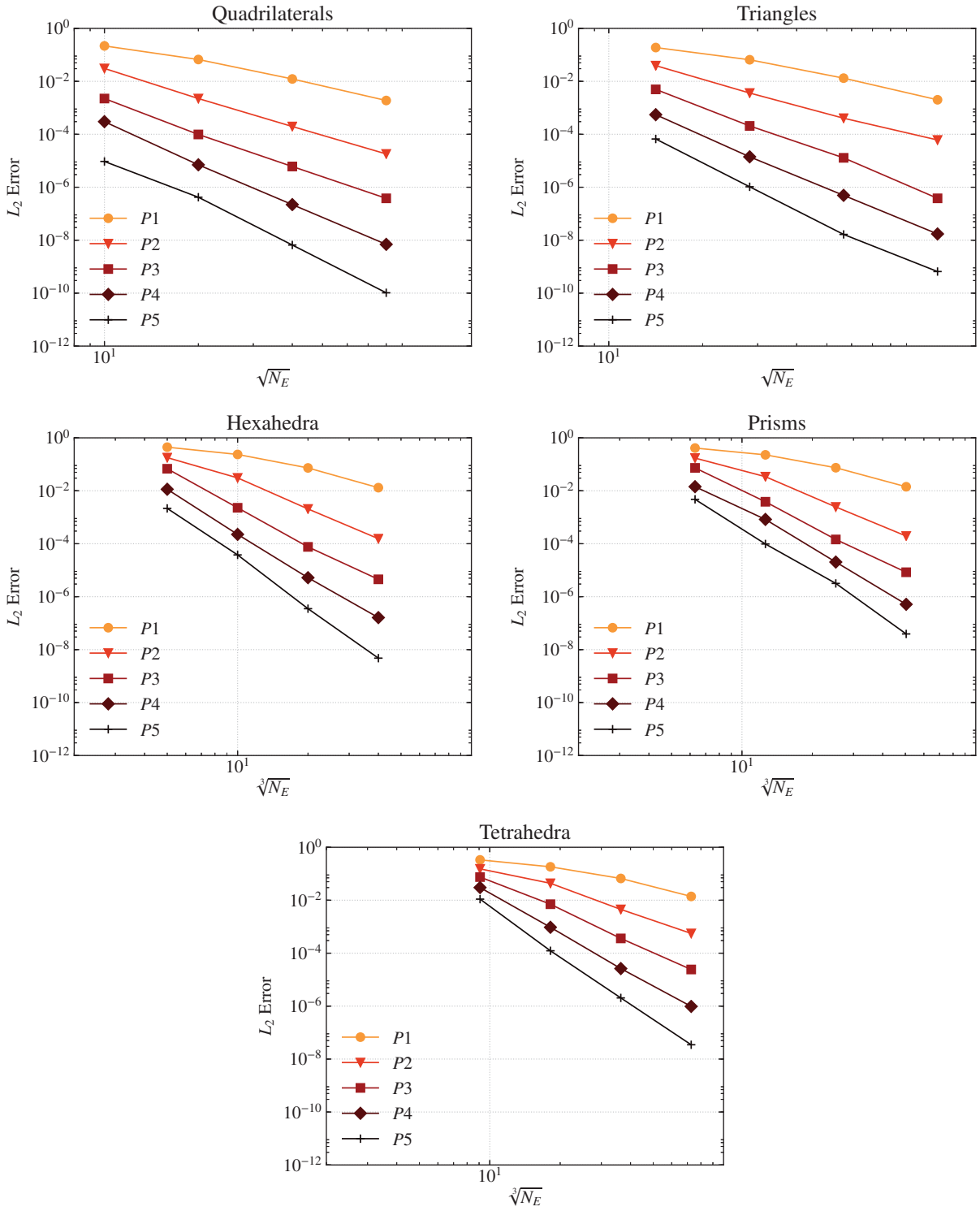


Figure 4.3. Grid convergence study of linear advection.

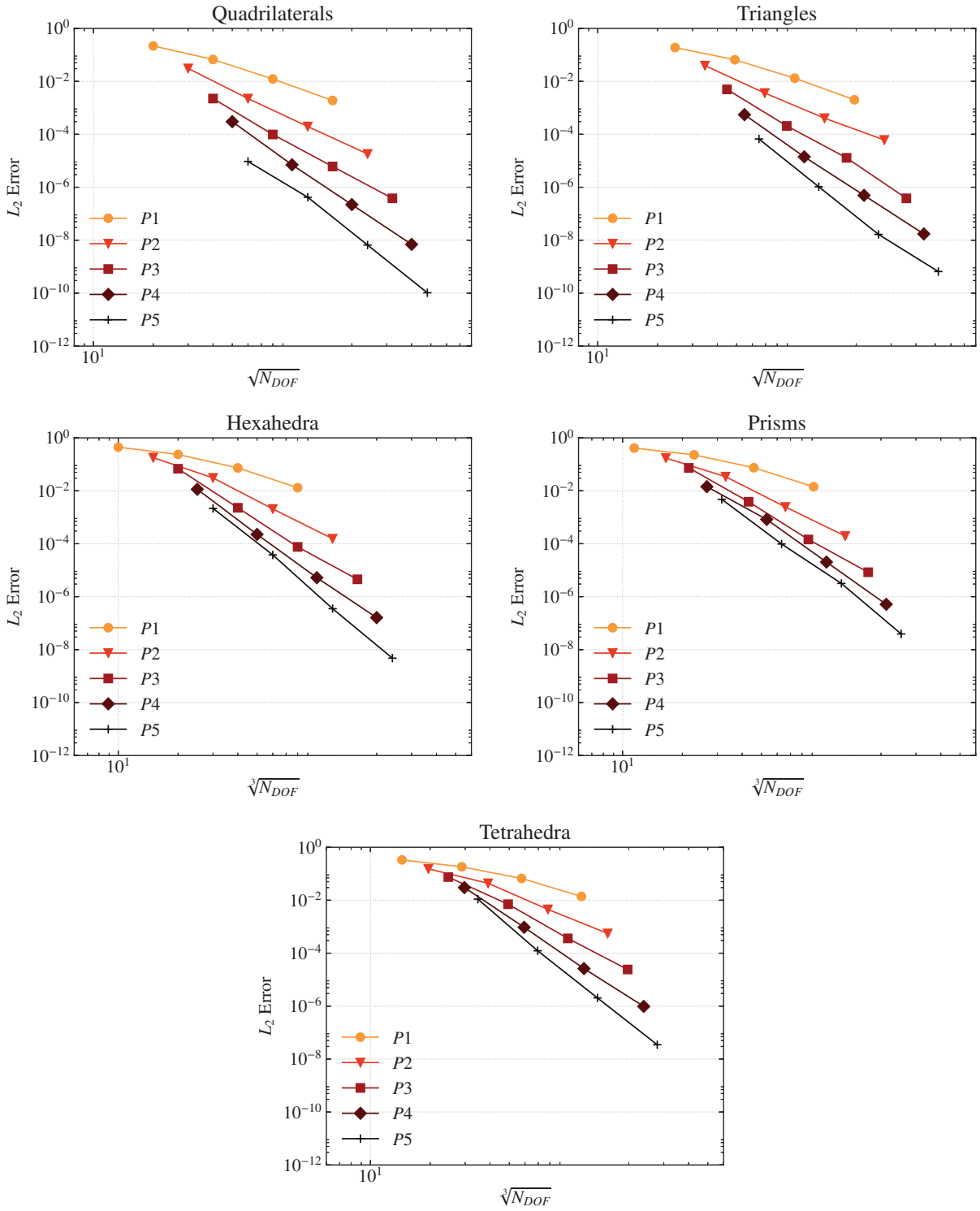


Figure 4.4. Accuracy of element types on a per degree of freedom basis for linear advection.

orders of accuracy, verifying the FR approach for multidimensional diffusion.

We analyze the accuracy of the element types for diffusion-dominated problems. Figure 4.6 displays the L_2 norm against the number of DOF for all considered element types. It is clear that both quadrilaterals and hexahedral elements are more accurate in two and three dimensions, respectively, compared to the other element types. This is more prominent for the finest grids and highest-order solution spaces P.

Table 4.3. L_2 norm of the solution error for two-dimensional linear diffusion.

		Quadrilaterals		Triangles	
P	N_c	L_2 Error	Order	L_2 Error	Order
1	5	$\mathbf{6.01} \times 10^{-2}$	-	$\mathbf{6.52} \times 10^{-2}$	-
	10	$\mathbf{1.80} \times 10^{-2}$	1.74	$\mathbf{2.07} \times 10^{-2}$	1.65
	20	$\mathbf{4.77} \times 10^{-3}$	1.92	$\mathbf{5.69} \times 10^{-3}$	1.87
	40	$\mathbf{1.21} \times 10^{-3}$	1.98	$\mathbf{1.47} \times 10^{-3}$	1.95
2	5	$\mathbf{6.15} \times 10^{-3}$	-	$\mathbf{1.02} \times 10^{-2}$	-
	10	$\mathbf{6.89} \times 10^{-4}$	3.16	$\mathbf{1.20} \times 10^{-3}$	3.08
	20	$\mathbf{8.34} \times 10^{-5}$	3.05	$\mathbf{1.46} \times 10^{-4}$	3.04
	40	$\mathbf{1.03} \times 10^{-5}$	3.01	$\mathbf{1.80} \times 10^{-5}$	3.01
3	5	$\mathbf{5.38} \times 10^{-4}$	-	$\mathbf{1.43} \times 10^{-3}$	-
	10	$\mathbf{3.66} \times 10^{-5}$	3.88	$\mathbf{9.71} \times 10^{-5}$	3.88
	20	$\mathbf{2.29} \times 10^{-6}$	4.00	$\mathbf{6.23} \times 10^{-6}$	3.96
	40	$\mathbf{1.43} \times 10^{-7}$	4.00	$\mathbf{3.93} \times 10^{-7}$	3.99
4	5	$\mathbf{2.97} \times 10^{-5}$	-	$\mathbf{1.68} \times 10^{-4}$	-
	10	$\mathbf{8.10} \times 10^{-7}$	5.20	$\mathbf{5.43} \times 10^{-6}$	4.95
	20	$\mathbf{2.43} \times 10^{-8}$	5.06	$\mathbf{1.72} \times 10^{-7}$	4.98
	40	$\mathbf{7.56} \times 10^{-10}$	5.01	$\mathbf{5.41} \times 10^{-9}$	4.99
5	5	$\mathbf{1.41} \times 10^{-6}$	-	$\mathbf{1.24} \times 10^{-5}$	-
	10	$\mathbf{2.35} \times 10^{-8}$	5.91	$\mathbf{2.18} \times 10^{-7}$	5.82
	20	$\mathbf{3.52} \times 10^{-10}$	6.06	$\mathbf{3.57} \times 10^{-9}$	5.93
	40	$\mathbf{5.37} \times 10^{-12}$	6.03	$\mathbf{5.69} \times 10^{-11}$	5.97

4.1.3 Planar Couette

Planar Couette flow is a well-known two-dimensional case to perform verification of the viscous fluxes given its simplification of the Navier-Stoke Equations. Couette flow consists of a viscous fluid between two plates separated with a distance of δ [50]. At $y = \delta$, there exists a moving wall with temperature T_e and constant velocity v_e, which drives the flow in the positive x-direction. At $y = 0$,

Table 4.4. L_2 norm of the solution error for three-dimensional linear diffusion.

		Hexahedra		Prisms		Tetrahedra	
P	N_c	L_2 Error	Order	L_2 Error	Order	L_2 Error	Order
1	4	6.7×10^{-2}	—	7.0×10^{-2}	—	7.3×10^{-2}	—
	8	2.1×10^{-2}	**1.68**	2.2×10^{-2}	**1.64**	2.5×10^{-2}	**1.55**
	16	5.6×10^{-3}	**1.90**	6.2×10^{-3}	**1.86**	7.3×10^{-3}	**1.78**
	32	1.4×10^{-3}	**1.98**	1.6×10^{-3}	**1.95**	1.9×10^{-3}	**1.92**
2	4	1.0×10^{-2}	—	1.5×10^{-2}	—	2.1×10^{-2}	—
	8	1.1×10^{-3}	**3.19**	1.8×10^{-3}	**3.06**	2.8×10^{-3}	**2.96**
	16	1.4×10^{-4}	**3.06**	2.1×10^{-4}	**3.05**	3.4×10^{-4}	**3.04**
	32	1.7×10^{-5}	**3.02**	2.6×10^{-5}	**3.02**	4.2×10^{-5}	**3.02**
3	4	1.0×10^{-3}	—	2.4×10^{-3}	—	4.7×10^{-3}	—
	8	7.6×10^{-5}	**3.77**	1.7×10^{-4}	**3.82**	3.3×10^{-4}	**3.84**
	16	4.8×10^{-6}	**3.97**	1.1×10^{-5}	**3.94**	2.1×10^{-5}	**3.95**
	32	3.0×10^{-7}	**4.00**	6.9×10^{-7}	**3.98**	1.3×10^{-6}	**3.99**
4	4	8.4×10^{-5}	—	3.4×10^{-4}	—	9.3×10^{-4}	—
	8	2.2×10^{-6}	**5.26**	1.1×10^{-5}	**4.93**	3.2×10^{-5}	**4.85**
	16	6.4×10^{-8}	**5.09**	3.6×10^{-7}	**4.96**	1.0×10^{-6}	**4.95**
	32	1.9×10^{-9}	**5.02**	1.1×10^{-8}	**5.01**	3.3×10^{-8}	**4.98**
5	4	4.1×10^{-6}	—	3.0×10^{-5}	—	1.4×10^{-4}	—
	8	7.7×10^{-8}	**5.74**	5.7×10^{-7}	**5.76**	2.6×10^{-6}	**5.77**
	16	1.1×10^{-9}	**6.05**	9.5×10^{-9}	**5.90**	4.3×10^{-8}	**5.92**
	32	1.7×10^{-1}	**6.04**	1.5×10^{-0}	**5.96**	6.9×10^{-0}	**5.97**

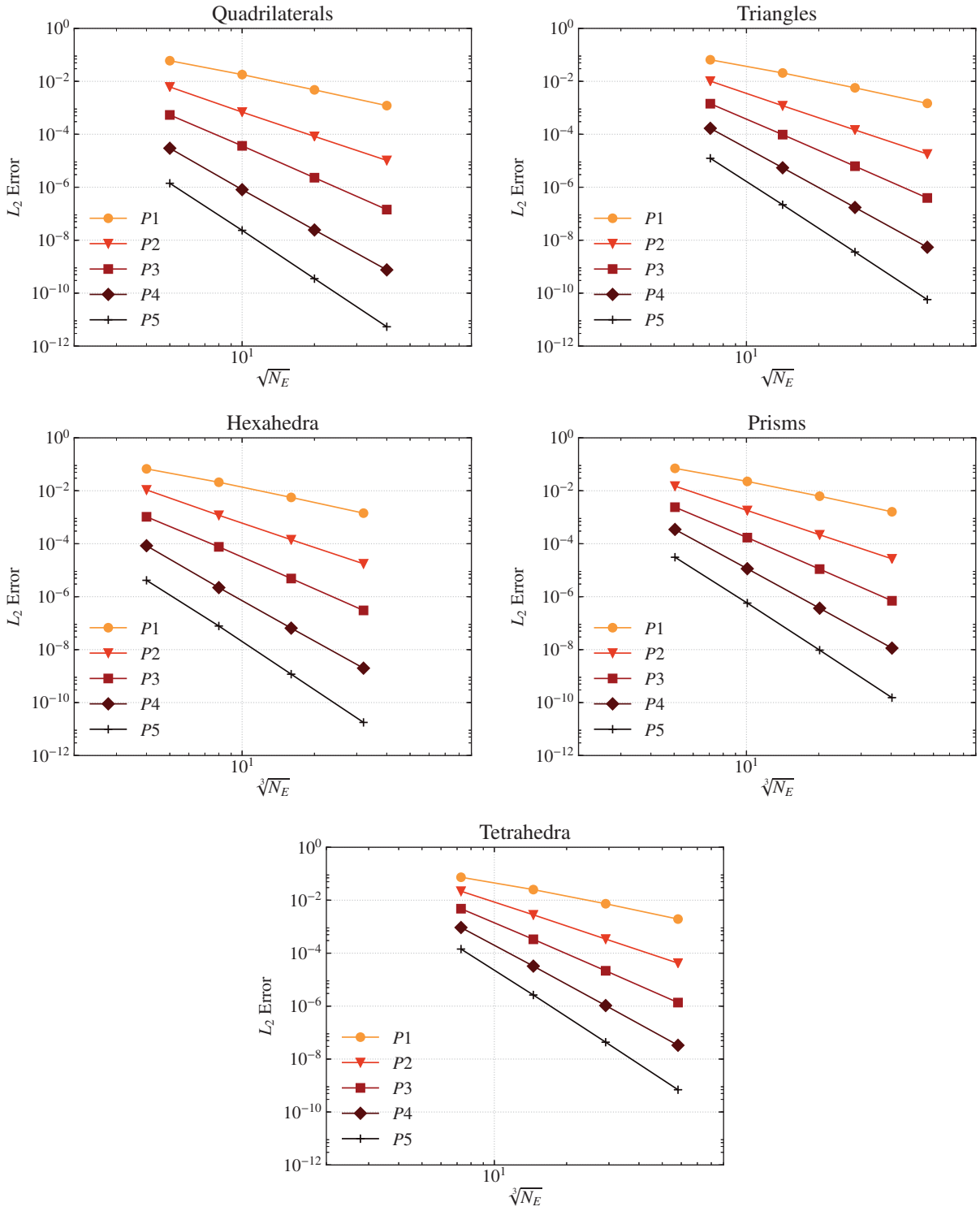


Figure 4.5. Grid convergence study of linear diffusion.

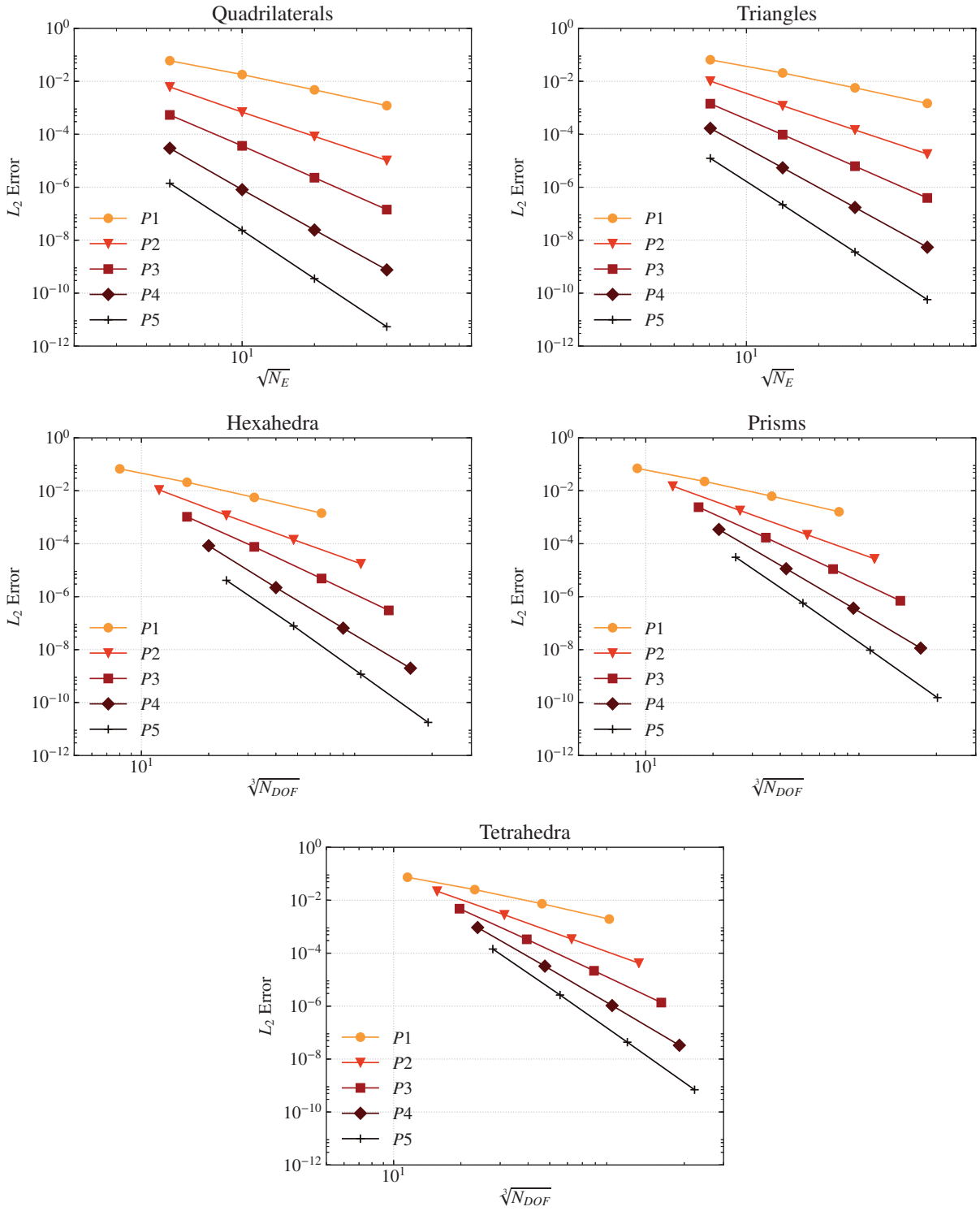


Figure 4.6. Accuracy of element types on a per degree of freedom basis for linear diffusion.

a fixed wall ($v_w = 0$) with temperature T_w is placed. Due to the no-slip condition, the flow variables are equal to those of the walls at $y = 0$ and $y = \delta$, respectively. Hence the flow experiences a temperature gradient due to viscous dissipation. The temperature profile can be computed from [50]

$$T = T_w + \left[T_e - T_w + \frac{Pr}{2c_p} v_e^2\right] \frac{y}{\delta} - \frac{Pr}{2c_p} v_e^2 \left(\frac{y}{\delta}\right)^2, \tag{4.6}$$

where $Pr = \mu \frac{c_p}{k}$ is the Prandtl number with μ the dynamic viscosity and k the thermal conductivity coefficient, c_p is the heat coefficient at constant pressure, y is the direction normal to the walls, and δ is the distance between the walls.

A rectangular computational domain is divided into $N_x = 4, 6, 8, 10$ elements and $N_y = \frac{N_x}{2}$. We perform the study using both quadrilaterals and triangular elements. The triangular elements are a simple subdivision of quadrilaterals into 2 triangles each, as shown in Figure 4.7. We implement Dirichlet boundary conditions for the flow variables at the walls and a constant viscosity μ. Similar to the diffusion case, the BR2 scheme is used for the viscous fluxes. The case is initialized with stationary flow and is allowed to advance in time until the solution converges to steady state using the fourth-order Runge-Kutta temporal scheme to minimize time discretization errors.

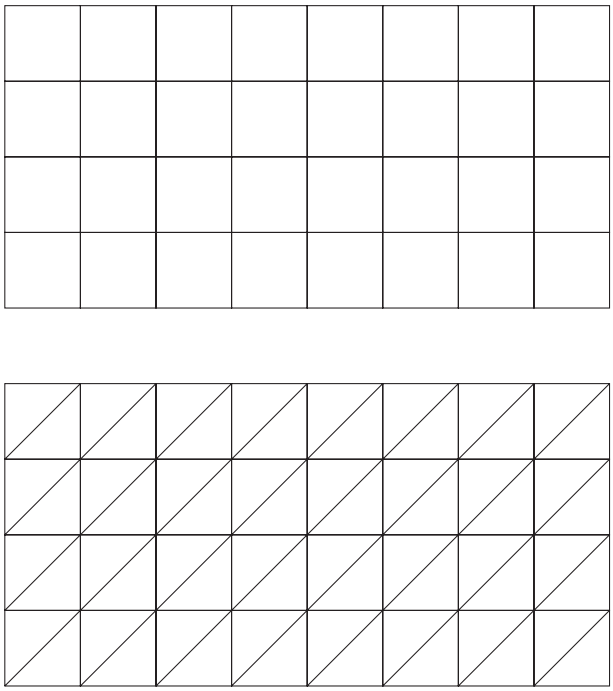


Figure 4.7. Quadrilateral and triangular grids for Couette flow.

Figure 4.8 shows contours of the temperature for the converged results using fourth and fifth-order spatial discretizations on triangular grids. Solution jumps are visible for $P3$ due to numerical error and the results improve with an increase in the polynomial degree.

The observed order of accuracy is evaluated using the L_2 norm from Equation 4.2 by comparing the predicted temperature T with the expected value. Figure 4.9 shows the convergence for both quadrilaterals and triangular elements. The orders of accuracy are consistent with the expected $P+1$ orders and can be further visualized in Table 4.5. Similar to the previous cases, the quadrilateral grids show more accurate results than those with triangles. This can be observed in Figure 4.10, where the error is plotted against the square root of the degrees of freedom.

(a) P3

(b) P4

Figure 4.8. Temperature profile of planar Couette flow using $P3$ and $P4$ schemes on triangular meshes.

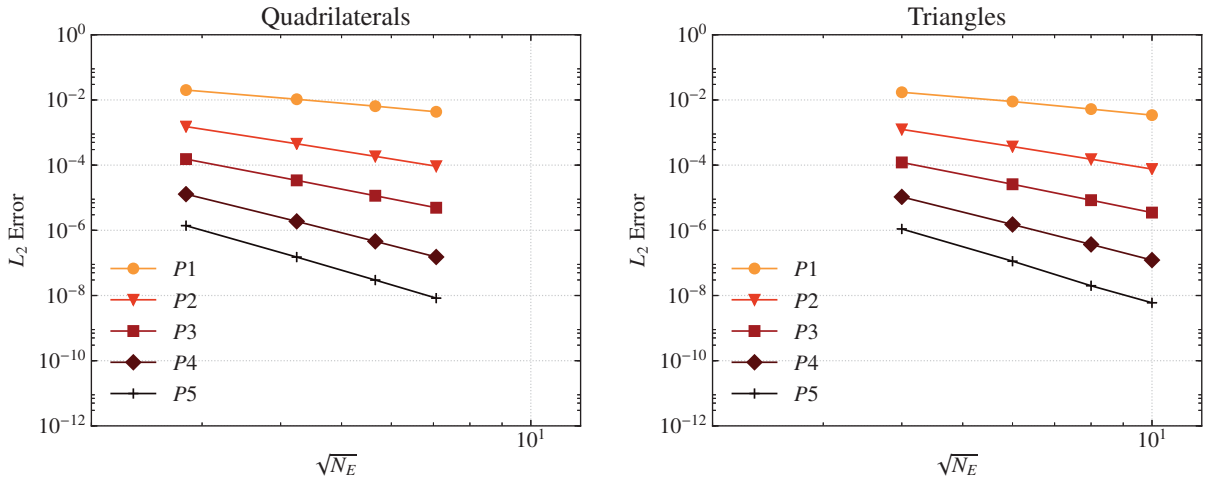


Figure 4.9. Grid convergence study of the L_2 norm of the temperature for Couette flow.

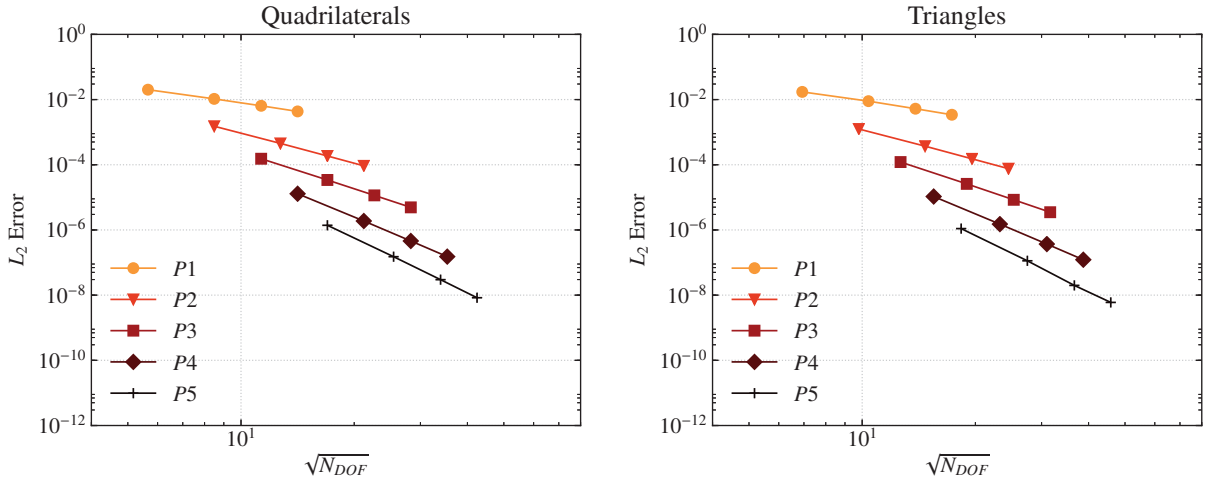


Figure 4.10. Accuracy of element types on a per degree of freedom basis for planar Couette flow.

Table 4.5. L_2 norm of the error for the planar Couette case.

		Quadrilaterals		Triangles	
P	N_c	L_2 Error	Order	L_2 Error	Order
1	4	2.01×10^{-2}	-	1.72×10^{-2}	-
	6	1.05×10^{-2}	**1.60**	8.94×10^{-3}	**1.62**
	8	6.44×10^{-3}	**1.70**	5.24×10^{-3}	**1.85**
	10	4.33×10^{-3}	**1.78**	3.42×10^{-3}	**1.92**
2 1	4	1.52×10^{-3}	-	1.25×10^{-3}	-
	6	4.52×10^{-4}	**2.99**	3.72×10^{-4}	**2.99**
	8	1.86×10^{-4}	**3.09**	1.52×10^{-4}	**3.12**
	10	9.27×10^{-5}	**3.11**	7.59×10^{-5}	**3.11**
3	4	1.53×10^{-4}	-	1.21×10^{-4}	-
	6	3.41×10^{-5}	**3.69**	2.59×10^{-5}	**3.81**
	8	1.15×10^{-5}	**3.77**	8.41×10^{-6}	**3.90**
	10	4.93×10^{-6}	**3.82**	3.50×10^{-6}	**3.93**
4	4	1.29×10^{-5}	-	1.06×10^{-5}	-
	6	1.88×10^{-6}	**4.75**	1.51×10^{-6}	**4.79**
	8	4.59×10^{-7}	**4.90**	3.69×10^{-7}	**4.90**
	10	1.52×10^{-7}	**4.96**	1.22×10^{-7}	**4.95**
5	4	1.39×10^{-6}	-	1.10×10^{-6}	-
	6	1.51×10^{-7}	**5.49**	1.13×10^{-7}	**5.59**
	8	2.98×10^{-8}	**5.63**	1.99×10^{-8}	**6.05**
	10	8.32×10^{-9}	**5.72**	5.98×10^{-9}	**5.39**

4.1.4 Isentropic Euler Vortex

The isentropic vortex consists of the passive convection of a vortex in a free-stream velocity field $\boldsymbol{v}_\infty = v_{x,\infty}\hat{\boldsymbol{\imath}} + v_{y,\infty}\hat{\boldsymbol{\jmath}}$. This test has been proposed by high-order workshops, such as [39], to test a solver's ability to preserve vorticity in an unsteady inviscid flow [39], which serves as verification of the Euler equations.

We perform the study on a two-dimensional square computational domain of length L with equal elements in x and y-directions $N_E = 10, 20, 40, 80$. The flow field conditions are initially specified as [41]

$$\rho = \left[1 - \frac{\beta^2 M_a^2 (\gamma - 1) e^{2f}}{8\pi^2}\right]^{\frac{1}{\gamma - 1}}, \tag{4.7}$$

$$v_x = \frac{\beta y e^f}{2\pi R}, \tag{4.8}$$

$$v_y = 1 - \frac{\beta x e^f}{2\pi R}, \tag{4.9}$$

$$p = \frac{\rho^\gamma}{\gamma {M_a}^2}, \tag{4.10}$$

where ρ is the density, β is the vortex strength, M_a is the Mach number, v_x and v_y are the x and y components of the velocity field, respectively, $f = \left(1 - x^2 - y^2\right)/2R^2$ and R is the radius of the vortex. We consider a radius of $R = 1.5$ and a vortex strength $\beta = 13.5$ similar to the work of Vermeire et al. [41].

The simulation is run for 20 convective times t^*, where $t^* = L/\|\boldsymbol{v}_\infty\|$, using temporal scheme RK_{44}. We perform the verification on quadrilaterals and triangular elements with second to sixth-order spatial discretizations. The error of the density solution is evaluated using the L_2 norm from equation 4.2. The observed orders of accuracy are proof of the benefits of using high-order methods and are consistent with the expected $P+1$ convergence order. Table 4.6 contains the detailed results for the grid convergence study.

For the sake of completeness, Figure 4.12 presents these results now plotted against the number of degrees of freedom per direction. Consistent with previous results, quadrilateral elements require smaller grids to advance a given level of error, compared to triangles.

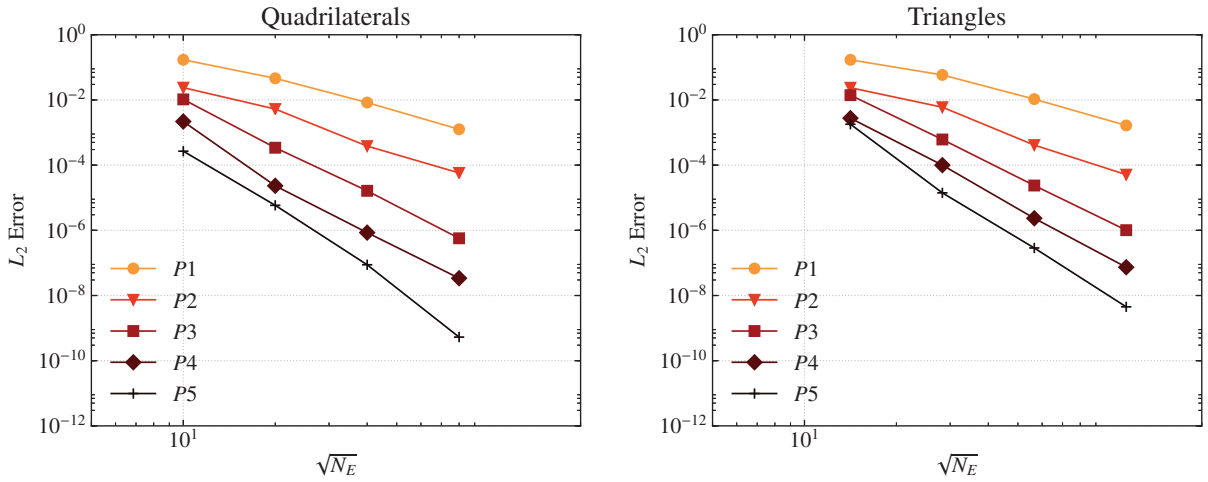


Figure 4.11. Grid convergence study of the L_2 norm of the density ρ for the isentropic vortex.

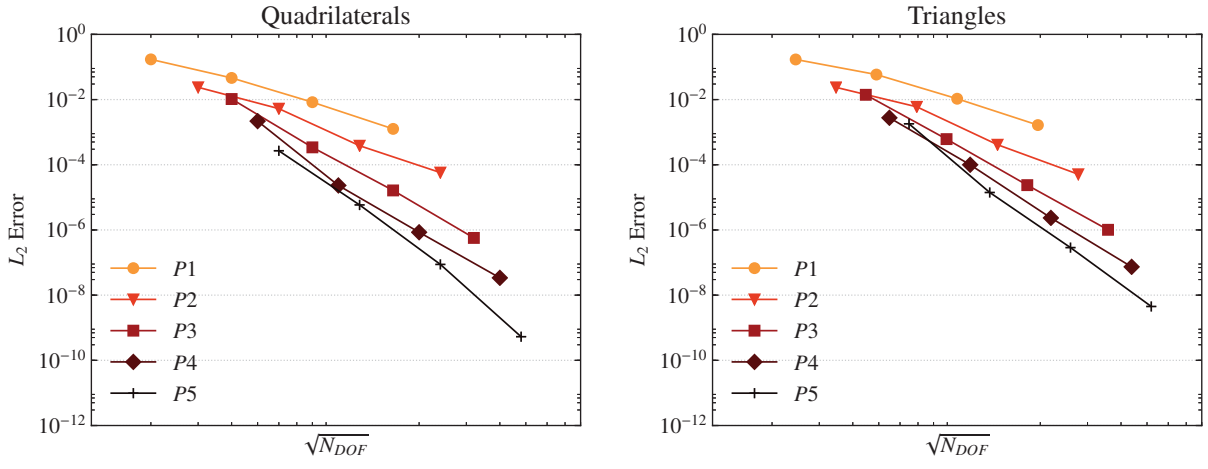


Figure 4.12. Accuracy of element types on a per degree of freedom basis for the isentropic vortex.

Table 4.6. L_2 norm of the density error for the isentropic vortex case.

		Quadrilaterals		Triangles	
P	N_c	L_2 Error	Order	L_2 Error	Order
1	10	1.71×10^{-1}	—	1.70×10^{-1}	—
	20	4.61×10^{-2}	**1.89**	5.85×10^{-2}	**1.54**
	40	8.31×10^{-3}	**2.47**	1.06×10^{-2}	**2.46**
	80	1.26×10^{-3}	**2.72**	1.65×10^{-3}	**2.69**
2	10	2.38×10^{-2}	—	2.37×10^{-2}	—
	20	5.26×10^{-3}	**2.18**	5.99×10^{-3}	**1.99**
	40	3.84×10^{-4}	**3.78**	4.12×10^{-4}	**3.86**
	80	5.76×10^{-5}	**2.74**	5.11×10^{-5}	**3.01**
3	10	1.04×10^{-2}	—	1.40×10^{-2}	—
	20	3.42×10^{-4}	**4.93**	6.12×10^{-4}	**4.52**
	40	1.63×10^{-5}	**4.40**	2.37×10^{-5}	**4.69**
	80	5.69×10^{-7}	**4.84**	1.01×10^{-6}	**4.55**
4	10	2.18×10^{-3}	—	2.76×10^{-3}	—
	20	2.32×10^{-5}	**6.56**	9.99×10^{-5}	**4.79**
	40	8.50×10^{-7}	**4.77**	2.34×10^{-6}	**5.42**
	80	3.39×10^{-8}	**4.65**	7.37×10^{-8}	**4.99**
5	10	2.68×10^{-4}	—	1.80×10^{-3}	—
	20	5.85×10^{-6}	**5.52**	1.41×10^{-5}	**7.00**
	40	8.77×10^{-8}	**6.06**	2.87×10^{-7}	**5.61**
	80	5.30×10^{-0}	**7.37**	4.49×10^{-9}	**6.00**

4.2 Temporal

A complete verification study in computational fluid dynamics evaluates also the implementation of the temporal schemes. In this section, we present a brief description of the fourth-order explicit RK method and two SDIRK schemes with second and third-order accuracy.

We verify the order of the temporal scheme using a simple linear advection case by subdividing a two-dimensional square computational domain into $N_x = N_y$ quadrilateral elements with an initial distribution of a scalar quantity $u(\boldsymbol{x}, t)$ from Equation 2.20

$$u(x, 0) = \prod_{i=1}^{N_{dim}} e^{-\frac{L}{10}(x_i - x_{i_c})^2}, \tag{4.11}$$

where x_{i_c} is the center of the domain along the i-th direction. For the two-dimensional case, $N_{dim} = 2$. We semidiscretize using a seventh-order scheme to minimize the spatial error and the solution was allowed to advect for a single convective time t^*. Figure 4.13 shows the convergence plots for the considered schemes, where the orders of accuracy are consistent with the order of the temporal schemes. Refer to Table 4.7 for further details. With these results, the temporal schemes are verified. We note that performing verification of explicit schemes require changing the grid spacing in conjunction with the time-step size to deal with the stability constraints.

In addition, we observe the size of the time steps used in the verification of each of these schemes. Implicit methods were able to converge using large time step sizes, while RK_{44} would have been unstable for those values.

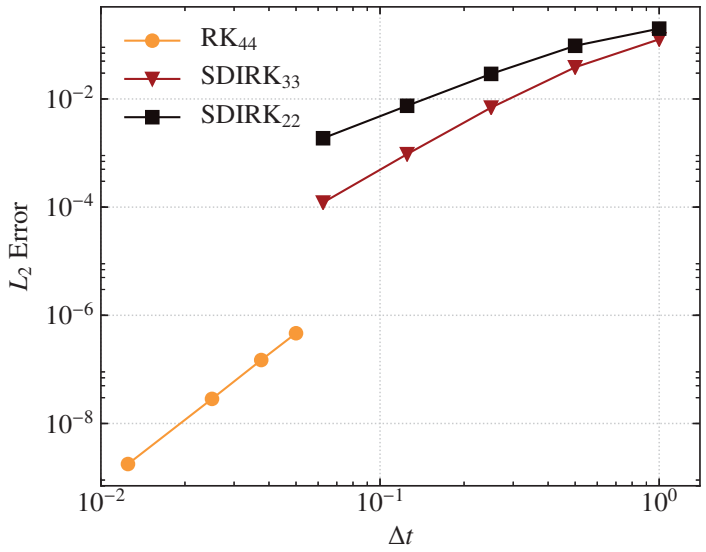


Figure 4.13. Verification of explicit and implicit time-marching schemes.

Table 4.7. L_2 norm of the solution error for the temporal verification.

Scheme	Δt	Error	Order
RK_{44}	0.05	4.66×10^{-7}	-
	0.0375	1.49×10^{-7}	**3.96**
	0.025	2.85×10^{-8}	**4.07**
	0.0125	1.78×10^{-9}	**4.00**
$SDIRK_{22}$	0.5	9.62×10^{-2}	-
	0.25	2.91×10^{-2}	**1.73**
	0.125	7.48×10^{-3}	**1.96**
	0.0625	1.88×10^{-3}	**1.99**
$SDIRK_{33}$	0.5	3.85×10^{-2}	**1.72**
	0.25	6.93×10^{-3}	**2.47**
	0.125	9.53×10^{-4}	**2.86**
	0.0625	1.22×10^{-4}	**2.97**

4.3 Comments

After performing a convergence study on the spatial and some temporal schemes in our in-house ILES solver, HORUS, we have verified their implementations. The observed orders of accuracy were generally consistent with the expected, $P + 1$ for the spatial FR schemes, and those of the temporal schemes. In addition, parallel to verification, a simple spatial accuracy analysis was presented to compare different two and three-dimensional element types. We found that tensor-product elements, such as quadrilaterals and hexahedra are the most accurate.

In the following chapter, we will analyze the numerical error of one and multi-dimensional semidiscretizations with tools from Fourier analysis. This study will provide more insights into why some elements are more accurate than others by looking in detail at the numerical error characteristic of each element type.

Chapter 5

Semidiscrete Analysis

A Flux-Reconstruction discretization is able to resolve turbulent structures up to a cut-off wavenumber, after which the numerical dissipation becomes significant. The amount of error introduced depends not only on the order of accuracy of the scheme but also on its spatial discretization, such as the common interface flux. Complex geometries benefit from more flexible element types such as tetrahedra and prisms, both of which are commonly used in unstructured grid generation. Hence, the importance of understanding the spectral properties of all element types, which is the focus of this work.

The construction of spectral curves allows us to explore the behaviour of numerical discretizations for a wide range of permissible wavenumbers. In the case of ILES at moderate to high Reynolds number, the compressible Navier-Stokes equations are dominated by their advection terms. Hence, previous studies have explored the spectral properties of linear advection for different element types [15, 18, 21]. Following this spectral analysis, the dispersive and dissipative characteristics of spatially discretized schemes in three-dimensions are evaluated for the Flux Reconstruction method and their impact on under-resolved turbulent scales is inferred.

In this chapter, we present the formulation of the spectral analysis of semidiscrete systems. We note that the focus of this work mainly regards three-dimensional spatial schemes. For the sake of completeness and to provide a general introduction to the analysis, we present the one-dimensional formulation and extend to multi-dimensions in the latter sections.

5.1 One-Dimensional Discretization

5.1.1 Formulation

Following Huynh's formulation [51], we present the procedure to analyze one-dimensional semidiscrete systems. In this sense, we evaluate the wave propagation behaviour in a semidiscrete system,

which are governed by linear advection

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0, \qquad F = \alpha u, \tag{5.1}$$

where u is the solution, $F = \alpha u$ is the advection flux, x is the spatial coordinate and t is time. Initially, we prescribe a harmonic plane wave solution

$$u(x,t) = e^{i(\kappa x - \nu t)}, \tag{5.2}$$

where κ defines the wavenumber, ν is the frequency and i is the imaginary number $\sqrt{-1}$. The exact solution can be easily determined by inserting the prescribed solution $u(x,t)$ into Equation 5.1, which results in the dispersion relation of Equation 5.1

$$\nu = \kappa\alpha. \tag{5.3}$$

We divide the computational domain Ω into N_E nonoverlapping elements Ω_e with length h_e. Consider equidistant elements with $h_e = h = 1$. Now, the solution must be projected to the Flux Reconstruction vector space, so it can be expressed as

$$u^\delta(x,t) = e^{i(\kappa x_e - \nu t)}\boldsymbol{U}, \tag{5.4}$$

where $\boldsymbol{U}$ is a vector that contains the amplitudes of the waves in vector space. Despite the prescription of a single Fourier component, the spatial discretization will represent it with several numerical polynomial modes. This causes the scheme to see a single wave as a combination of Fourier modes, each with its own apparent frequency [16].

Recall from Chapter 3 the resulting equation for the flux derivative in Equation 3.17, which we rewrite with all the terms expanded

$$(\boldsymbol{F}_e)_x = \frac{2}{h_e}\left(D\boldsymbol{f}_e^\delta + \left[f_{e,L}^* - F_e^\delta(-1)\right]\boldsymbol{g}_{\xi,L} + \left[f_{e,R}^* - F_e^\delta(1)\right]\boldsymbol{g}_{\xi,R}\right). \tag{5.5}$$

To account for the jumps at the interfaces, we must define a Riemann solver. In this work, we make

use of the upwind scheme. Assuming information travels in the positive direction ($\alpha > 0$)

$$f^*_{e,L} = f_{e-1,R}, \qquad f^*_{e,R} = f_{e,R}, \tag{5.6}$$

where f^* denotes the common flux at the interfaces. These can be readily computed by interpolation when the solution points are not located at the interfaces. Hence,

$$f^*_{e,L} = \boldsymbol{\phi}(1)\boldsymbol{f}_{e-1} = \boldsymbol{r}^T\boldsymbol{f}_{e-1}, \tag{5.7}$$

$$f^*_{e,R} = \boldsymbol{\phi}(-1)\boldsymbol{f}_{e} = \boldsymbol{l}^T\boldsymbol{f}_{e}. \tag{5.8}$$

For linear advection, we rewrite Equation 5.5

$$(\boldsymbol{F}_e)_x = \frac{2}{h_e}\alpha\left(D\boldsymbol{u}^\delta_e + \left[\boldsymbol{r}^T\boldsymbol{u}^\delta_{e-1} - \boldsymbol{l}^T\boldsymbol{u}^\delta_e\right]\boldsymbol{g}_{\xi,L}\right). \tag{5.9}$$

Note that $f^*_{e,R} = F^\delta_e(1)$ so the right correction term in Equation 5.5 vanishes. Next, the solution at the neighbouring elements can be found by a switch in the index [9]

$$\boldsymbol{u}^\delta_{e-1} = e^{-i\kappa x_e}\boldsymbol{u}^\delta_e. \tag{5.10}$$

By grouping elements with similar coefficients and substituting in Equation 5.9, we obtain

$$i\frac{\nu h}{\alpha}\boldsymbol{U} = M\boldsymbol{U}, \tag{5.11}$$

where M is the semidiscrete matrix given by

$$M = 2\left(\left[D - \boldsymbol{l}^T\boldsymbol{g}_L\right] + \boldsymbol{r}^T\boldsymbol{g}_{\xi,L}e^{-i\kappa x_e}\right), \tag{5.12}$$

and is commonly written in terms of the linear transformation Jacobian blocks C. Hence,

$$M = 2\left(C_0 + C_{-1}e^{-i\kappa x_e}\right), \tag{5.13}$$

where $C_0 = D - \boldsymbol{l}^T\boldsymbol{g}_L$ and $C_{-1} = \boldsymbol{r}^T\boldsymbol{g}_L$ are the diagonal and off-diagonal block entries of the Jacobian

matrix. Equation 5.11 is a classical eigenvalue problem with $P+1$ complex solutions $\boldsymbol{\lambda}$ that define numerical frequencies $\boldsymbol{\omega}^{\delta}$ by [15]

$$\omega_j^{\delta} = \frac{\alpha \lambda_j}{ih}. \tag{5.14}$$

The complex characteristic of the numerical frequency establishes the presence of numerical dissipation. By looking at the prescribed solution in Equation 5.4, we observe that the imaginary component of a given numerical frequency ω_j^{δ} would act as an amplification modifier of the initially prescribed wave. The real part of ω_j^{δ}, when different from $\alpha\kappa$, would cause the jth mode to travel at incorrect speed, introducing dispersive behaviour to the final solution. In addition, none of the Fourier modes are allowed to grow in time. Hence, the imaginary part of the numerical frequency must only contain nonpositive imaginary values so the numerical error stays bounded.

5.1.2 Eigenspectra

We can gain some general insights on the numerical stability of the spatial scheme by looking at the collection of eigenvalues of the semidiscrete matrix. When coupling a spatial scheme with explicit time-marching methods, the eigenvalues are scaled by the time-step size Δt to fit within their stability regions. This means that semidiscrete systems with larger eigenspectra require smaller time-step sizes [21]. As shown in Figure 5.1, the region of eigenvalues pertaining to higher-order schemes is larger. This explains why higher-order methods require a smaller time-step size when using explicit temporal schemes. The size of the eigenspectra measured by the minimum real value is shown in Table 5.1. As we observe, the size of these regions grows faster than the polynomial degree.

Table 5.1. Minimum eigenvalues on the real axis for $P0$ to $P6$ one-dimensional semidiscretizations.

P	$\min(\Re(\lambda))$
0	-2.000
1	-5.999
2	-11.842
3	-19.155
4	-27.842
5	-37.821
6	-49.052

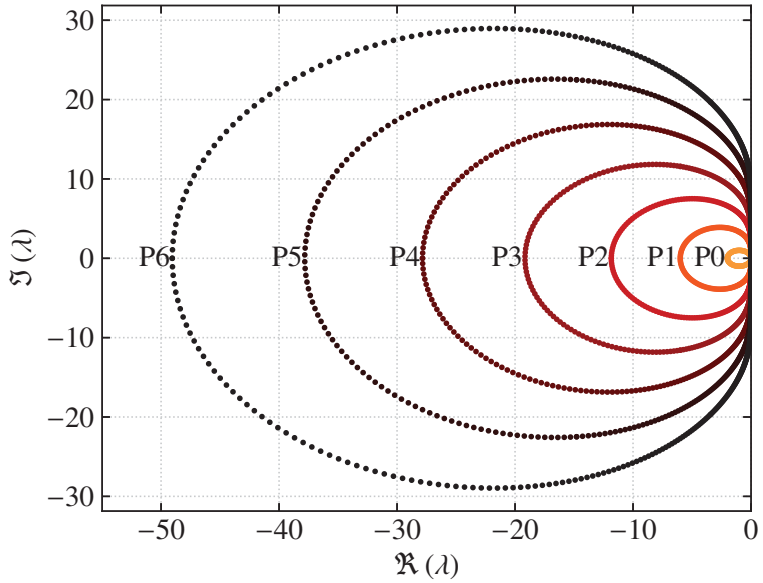


Figure 5.1. Collection of eigenvalues for $P0$ to $P6$ one-dimensional FR schemes.

5.1.3 Dispersion Relations

The study of the numerical error of one-dimensional schemes has been carried out by several authors [15, 20, 21, 23] to understand the stability and dominant type of numerical error of high-order schemes. We present a general review of the findings by these authors to understand the spectral properties of high-order methods.

Figures 5.2 and 5.3 display the numerical dispersion relations of one-dimensional spatial discretizations for $P0$ to $P6$ schemes. The resulting relations show the amount of numerical error introduced by the scheme. Dispersion curves are shown on the left side and dissipation on the right side. Clearly, the range of fully resolved wavenumber increases with the order of the scheme. This means that, for a given grid spacing, we introduce less numerical dispersion and dissipation with higher P. Moura [17] observed that when dissipation effects are greater than 1%, the accuracy of the solution began to drop. After this point, the numerical scheme acts as a low-pass filter and significantly damps the higher frequencies. This numerical dissipation acts in lieu of a subgrid-scale turbulence model and provides stabilization for ILES simulations [14].

High-order schemes are characterized by a cut-off wavenumber defined by $\kappa_c = (P+1)\pi$, which is the maximum wavenumber that can be spatially represented given the number of degrees of freedom.

However, we note that κ_c is an overestimation of the capabilities of an FR scheme since the numerical error has completely dominated the solution before this point. Furthermore, Figures 5.2 and 5.3 show a distinction between a dark curve and several others in light color. The dark curve represents the dominant physical mode. The eigenvalues that compose this curve dominate the numerical error in the solution at low to medium wavenumbers [15]. On the other hand, the light curves represent secondary modes, which have been shown to have contributions to the final result. These modes are a replica of the physical mode every 2π due to aliasing error. We present more on the influence of secondary modes in a later section for higher spatial dimensions by looking at the eigenfunctions of the semidiscretizations.

To fairly compare the properties of numerical schemes with the exact dispersion relation, Figure 5.4 shows the dominant modes of each P-scheme against the nondimensional wavenumber $\bar{\kappa} = kh/(P+1)$. We observe the higher-order schemes follow the exact dispersion relation for the higher ranges of wavenumbers compared to low-order methods, which means that on equal grids, high-order methods will be more accurate. In general, it is the dissipation effects that predominantly characterize upwinded Flux Reconstruction schemes [9].

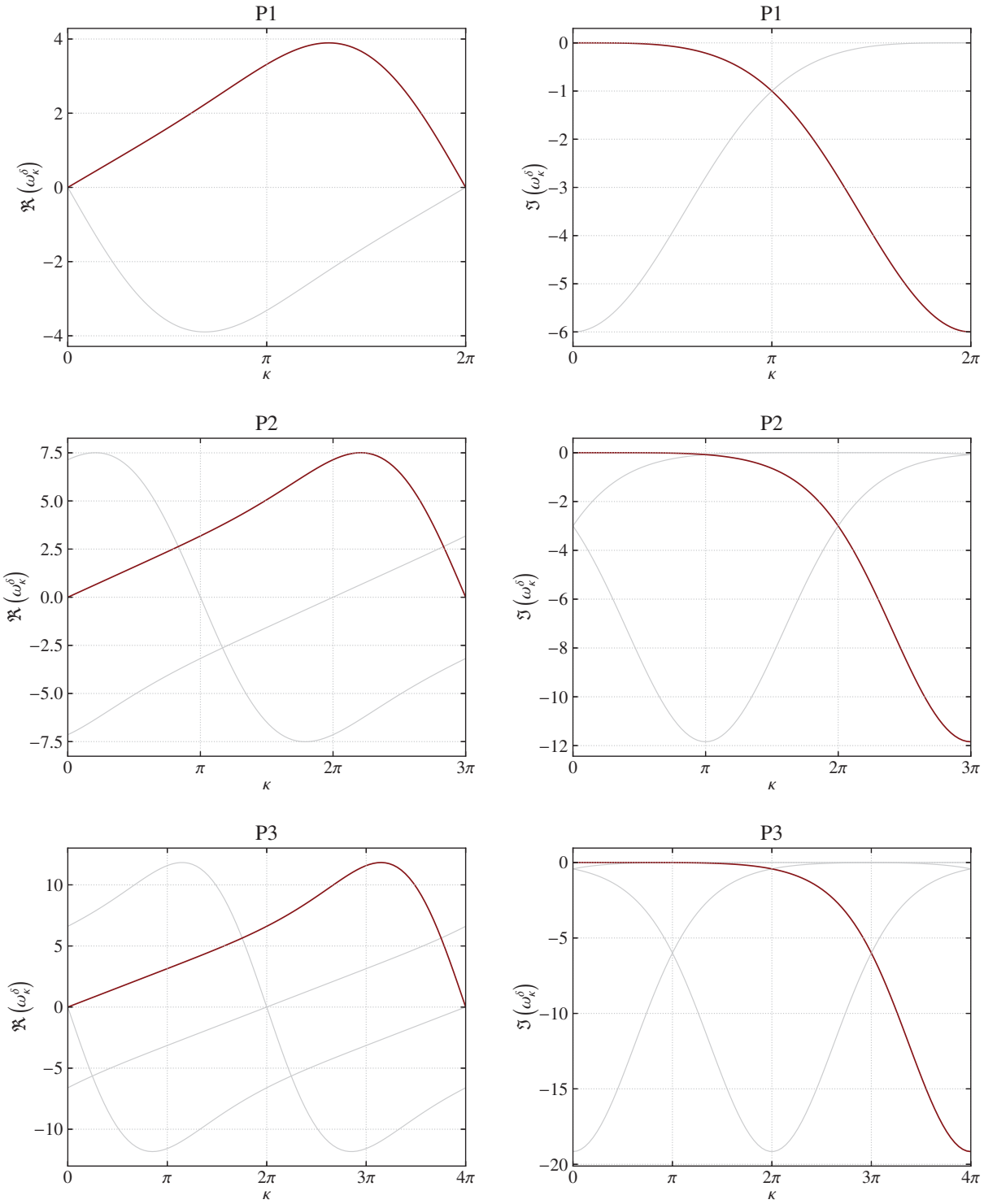


Figure 5.2. Dispersion relations for the one-dimensional case using $P1$ to $P3$ schemes.

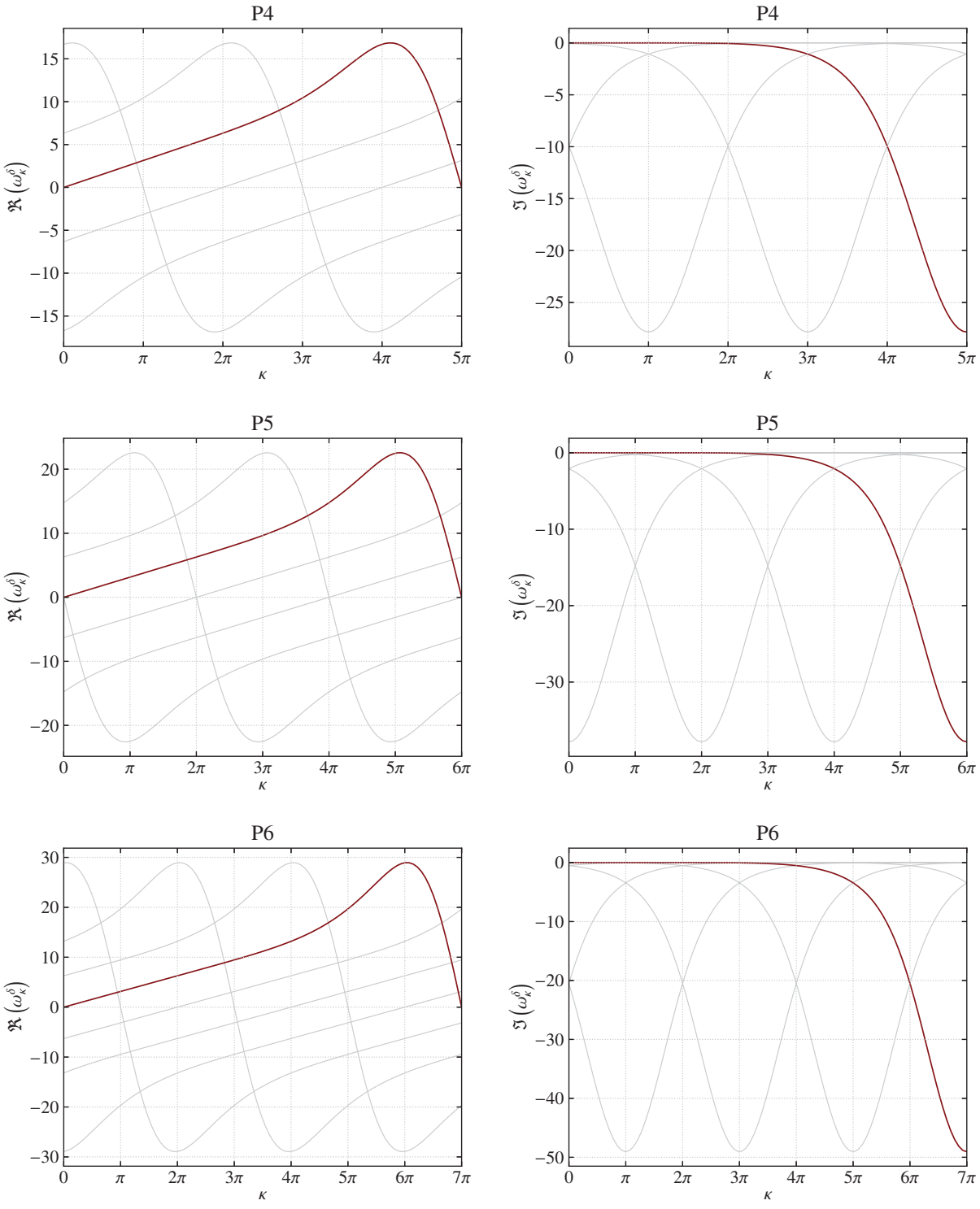


Figure 5.3. Dispersion relations for the one-dimensional case using $P4$ to $P6$ schemes.

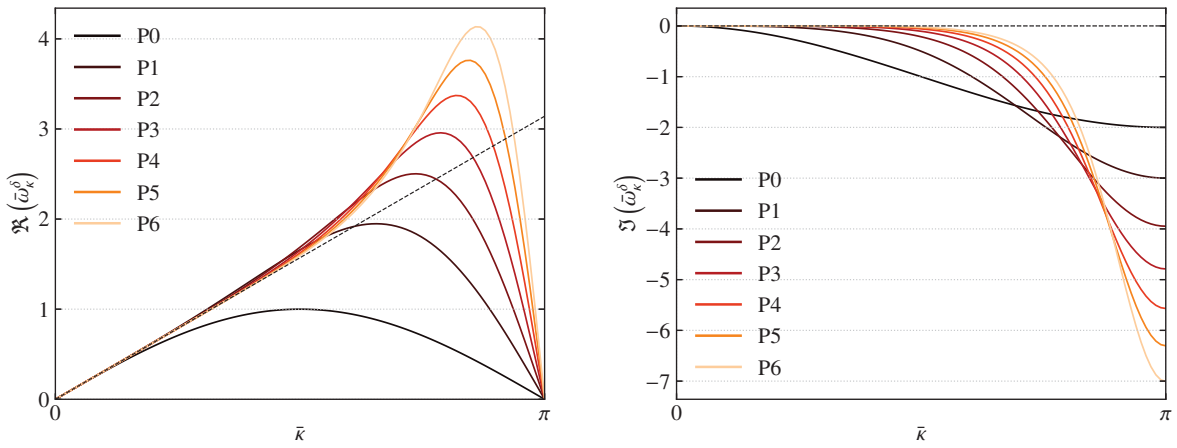


Figure 5.4. Comparison of numerical error for $P0$ to $P6$ FR schemes.

5.2 Multidimensional Discretizations

5.2.1 Formulation

Consider the multidimensional linear advection equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{\alpha} u) = 0, \tag{5.15}$$

where $u = u(\boldsymbol{x}, t)$ is the solution variable, $\boldsymbol{\alpha} u$ is the linear advection flux and t is time. The propagation vector $\boldsymbol{\alpha} = \boldsymbol{\alpha}(\theta_1, \theta_2)$ can defined by

$$\boldsymbol{\alpha} = \|\boldsymbol{\alpha}\| \mathbf{1}_\alpha, \tag{5.16}$$

where

$$\mathbf{1}_\alpha(\theta_1, \theta_2) = \left[\cos(\theta_1)\cos(\theta_2), \sin(\theta_1), \cos(\theta_1)\sin(\theta_2)\right]^T, \tag{5.17}$$

and $\|\cdot\|$ denotes the magnitude of the vector as shown in Figure 5.5.

Following [15, 18, 21], we admit plane wave solutions of the form

$$u = e^{i(\boldsymbol{\kappa}\cdot\boldsymbol{x} - \nu t)}, \tag{5.18}$$

where $\boldsymbol{\kappa} = \boldsymbol{\kappa}(\beta_1, \beta_2) = \|\boldsymbol{\kappa}\| \mathbf{1}_\kappa$ is the wave vector, $\boldsymbol{x} = [x_1, x_2, x_3]^T$, ν is exact the frequency, $\boldsymbol{x}$ is

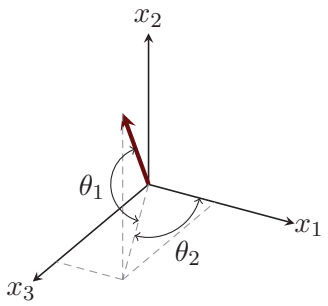


Figure 5.5. Three-dimensional vector decomposition reference.

the physical coordinate and i is the imaginary number $\sqrt{-1}$. A periodic computational domain in $\boldsymbol{x}$ is then split into N_c nonoverlaping rectangular cuboids Ω_c, which can be further split into $N_{sub} = 2$ prismatic or $N_{sub} = 6$ tetrahedral volumes for a total of $N_e = N_{sub}N_c$ elements (see Figure 4.1. The cuboid blocks have edges of length h. For simplicity, we consider h to be unity.

The admitted solution must then be projected to the FR space to comply with the vector characteristic form of the scheme. This yields

$$\boldsymbol{u}_e^{\delta} = e^{i(\boldsymbol{\kappa}\boldsymbol{x}_c - \nu t)}\boldsymbol{U}, \tag{5.19}$$

with $\boldsymbol{\kappa}$ the prescribed wave vector, $\boldsymbol{x}_c$ the centroid position of the rectangular cuboid and $\boldsymbol{U}$ a vector of unknowns independent of the solution points. Discretizing Equation 5.15 in space using the FR scheme with an upwind Riemann solver and the DG correction field from Equation 3.33 yields an expression of the form

$$\frac{d\boldsymbol{U}}{dt} = M\boldsymbol{U}, \tag{5.20}$$

where for a given element type M is the square semidiscrete matrix defined by the propagation velocity $\boldsymbol{\alpha}$, mesh geometry, wavevector $\boldsymbol{\kappa}$, Riemann solver and penalty term η (see Chapter 3). It is the result of the dot product between vector $\boldsymbol{C}$, which contains the linear transformation operator matrices, each of size $m = N_{sub}N_{DOF}$, and the vector $\boldsymbol{E}$ with entries from the neighbouring elements, as shown in Figure 5.6. Note that the first entry of vector $\boldsymbol{E}$ is one, referring to the solution in Ω_c. At least three rectangular cuboids are required in each direction to achieve computational periodicity, since the FR approach has a stencil including only one neighbouring element on each face. In three

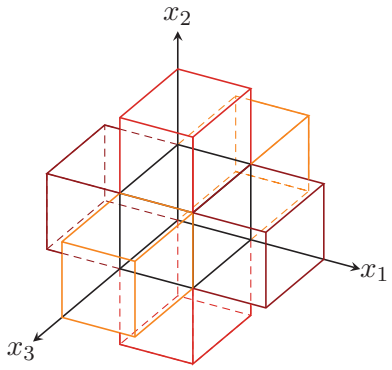


Figure 5.6. Cuboidal periodic pattern of a 3D mesh. Neighbouring elements are shown in red (darkest) for the $\hat{\imath}$-direction, orange (medium) for the $\hat{\jmath}$-direction, and yellow (light) for the $\hat{k}$-direction.

dimensions this yields

$$\boldsymbol{C} = \left[C^{0,0,0}, C^{-1,0,0}, C^{+1,0,0}, C^{0,-1,0}, C^{0,+1,0}, C^{0,0,-1}, C^{0,0,+1}\right]^T, \tag{5.21}$$

$$\boldsymbol{E} = \left[1, e^{-i\kappa\hat{\imath}}, e^{+i\kappa\hat{\imath}}, e^{-i\kappa\hat{\jmath}}, e^{+i\kappa\hat{\jmath}}, e^{-i\kappa\hat{k}}, e^{+i\kappa\hat{k}}\right], \tag{5.22}$$

$$M = \frac{\|\boldsymbol{\alpha}\|}{h}\boldsymbol{E}\cdot\boldsymbol{C}, \tag{5.23}$$

and from Equation 5.20 we obtain

$$i\frac{\nu h}{\|\boldsymbol{\alpha}\|}\boldsymbol{U} = M\boldsymbol{U}. \tag{5.24}$$

Equation 5.24 is a classical eigenvalue problem with complex eigenvalues $\boldsymbol{\lambda}$ that yield numerical frequencies $\boldsymbol{\omega}^\delta = \alpha\boldsymbol{\lambda}/ih$, whose components are $\omega_j^\delta,\ j = 1, \ldots, m$. Several analyses [15, 18, 21] have shown that there exists a physical mode which most closely represents the Fourier mode $e^{i\boldsymbol{\kappa}\cdot\boldsymbol{x}}$ at low wavenumbers. The $m-1$ so-called spurious modes contribute to the final solution, and can dominate at relatively high wavenumbers. Further discussion of this follows in the next subsection. Hence, by substituting the plane wave solution in Equation 5.15, we obtain the exact dispersion relation $\nu = \boldsymbol{\alpha}\cdot\boldsymbol{\kappa}$. For any prescribed value of $\boldsymbol{\kappa}$, we expect

$$\Im\left(\omega_j^\delta\right) = 0, \quad \Re\left(\omega_j^\delta\right) = \|\boldsymbol{\kappa}\|. \tag{5.25}$$

Any deviation from these relations is an indication of error introduced to the numerical solution. In addition, the imaginary part of ω_j^δ must be nonpositive to ensure boundedness and is associated

with the loss of energy of the initial plane wave, whereas $\Re(\omega^\delta) \neq \|\boldsymbol{\alpha} \cdot \boldsymbol{\kappa}\|$ is associated with phase shift incongruence.

5.2.2 Analysis of Dominant and Secondary Modes

The computational cost of propagating wave-like solutions is proportional to the number of degrees of freedom within each rectangular cuboid. Cuboids composed of prismatic and tetrahedral elements with total degrees of freedom $N_{sub}N_{DOF}$ (see Table 3.1) have an advantage due to their subdivision over the hexahedral cuboid: the cuboidal ensemble of $N_{sub} > 1$ contains a higher number of DOF in comparison with the hexahedra having $N_{sub} = 1$. Therefore, in order to establish a fair comparison in terms of computational cost, consider the ratio of degrees of freedom within the cuboid $\gamma_{c,e} = N_{sub}N_{DOF_e}/N_{DOF_{hex}}$ in which the DOF are normalized with the geometrically simple hexahedral elements. Hence, all plots in this section make use of the nondimensional wavenumber $\bar{\kappa} = \frac{\kappa h}{\gamma_{c,e}}$. We acknowledge that the computational performance does not solely depend on the number of DOF, but a more fair element comparison can be reached with $\gamma_{c,e}$.

The meaning of all the m modes $\boldsymbol{\omega}^\delta$ in the solution has been previously discussed by Moura [15]. In one dimension, $m-1$ of the m modes have been viewed as non-physical components that contribute to the final result. In higher dimensions, Van den Abeele [21] identified more than one physical mode in triangular meshes. A fundamental or physical mode is most of the time one which contains the maximum energy for a given wavenumber. While more than one mode may become the most energized as the wavenumber increases, generally a single eigenvalue ω_j^δ dominates the dispersion and dissipation behaviour of the spatial discretization, while the other numerical frequencies contain large imaginary components at the region of fully resolved Fourier modes. In this sense, we consider the effects of the mode which is least dissipative for the largest range of wavenumbers, and we will refer to it as the dominant mode.

Previous research has characterized the shape of the eigencurves for one and two dimensions, where the physical modes are replicated as the wavenumber increases, as shown in Section 5.1.3. Such modal replication has been associated with aliasing effects [16]. Nevertheless, a fully periodic behaviour is not always observed in three dimensions, in particular when the wave and propagation orientations do not coincide with the geometry of the grid or its diagonal directions. Furthermore, upwinded schemes have generally displayed continuous eigencurves, and it is the upwinding factor

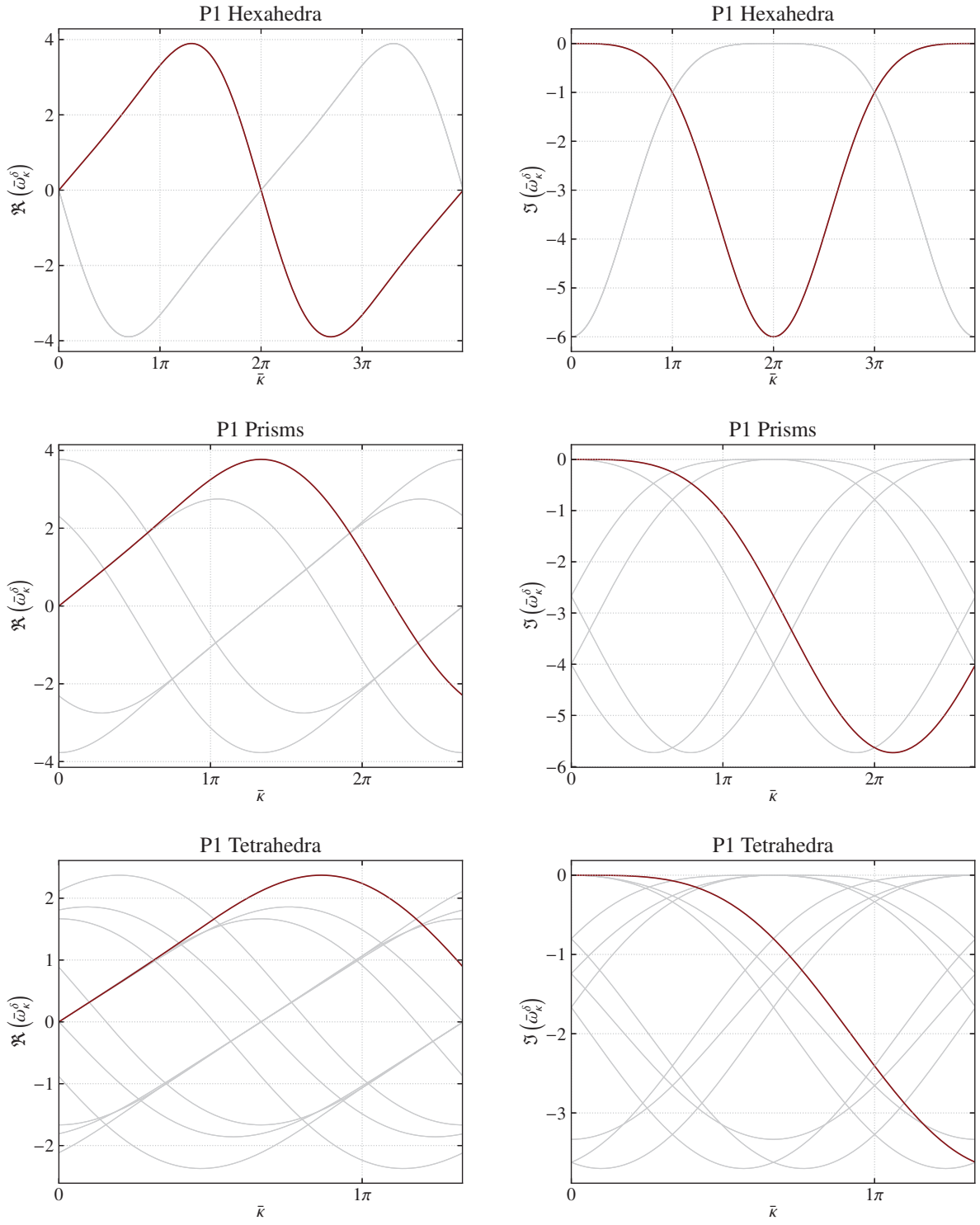


Figure 5.7. Eigenmodes for a wave aligned with the x_1 direction on three-dimensional $P1$ schemes.

which has been associated with dissipation error. Generally, the introduction of the third spatial coordinate results in schemes with comparable properties to lower dimensional scenarios for given wave orientations. Nevertheless, the pattern of the curves may differ and new characteristics appear under more general configurations. To visualize this, we present some discussion on the numerical dispersion and dissipation properties for certain wavevector directions: aligned with the grid, oriented in the direction of the main diagonal of the cuboid, and an arbitrarily chosen set of angles.

Figure 5.7 shows the spectral properties of all element types for a wavevector aligned with the x_1 direction. For visualization purposes, only results for the second-order scheme are shown, given that higher-order plots become significantly dense. We refer the reader to Appendix B for these plots with $P > 1$. Note that the general behaviour was observed to be similar with higher P, with the obvious increases in the well-resolved Fourier mode range. When recovering a quasi-one-dimensional case ($\theta_1 = \theta_2 = 0$), hexahedral elements are able to recover the exact one-dimensional dispersion and dissipation properties. The $m = (P+1)^3$ modes are equally concentrated into only $(P+1)$ curves. So, in the case of $P = 1$, only two curves can be observed. This is a consequence of the resulting block diagonal spatial matrix M with block entries recovering the one-dimensional semidiscrete spatial operator. Thus, its eigenvalues are exactly those of the one-dimensional case.

Contrary to the hexahedra, the curves of the prisms and tetrahedra do not have overlapping modes. The m modes can be found within the range $\bar{\kappa} \in \left[0, \frac{2\pi}{\gamma_e}\right]$. In addition, the dispersion curves do not evolve symmetrically over the zero dissipation axis as the wavenumber is increased. This is observed in the figures for prisms and tetrahedra, where the negative dispersion reaches different maxima and minima. Nevertheless, the curves are clearly a repetition of the dominant mode, highlighted in Figure 5.7. All spectral information is contained within $[0, (P+1)\,\pi]$ for this configuration. Hence, the cuboid of subdivision $N_{sub} > 1$ is not able to recover the one-dimensional case.

Changing the wavevector orientation to a diagonal direction, such as the main diagonal of the cuboid with $\theta_2 = \frac{\pi}{4}$ and $\theta_1 = \tan^{-1}\left(\cos\left(\frac{\pi}{4}\right)\right)$ results in constantly-replicated modes in spectral space, as observed in Figure 5.8. The shape of these curves is however quite interesting, in particular for the tetrahedral and prismatic elements. Despite the implementation of a fully upwinded Riemann Solver, jumps in the eigenmodes can be observed when considering prisms or tetrahedra. This behaviour is typical of spatial discretizations with central fluxes and has been also observed for upwind cases

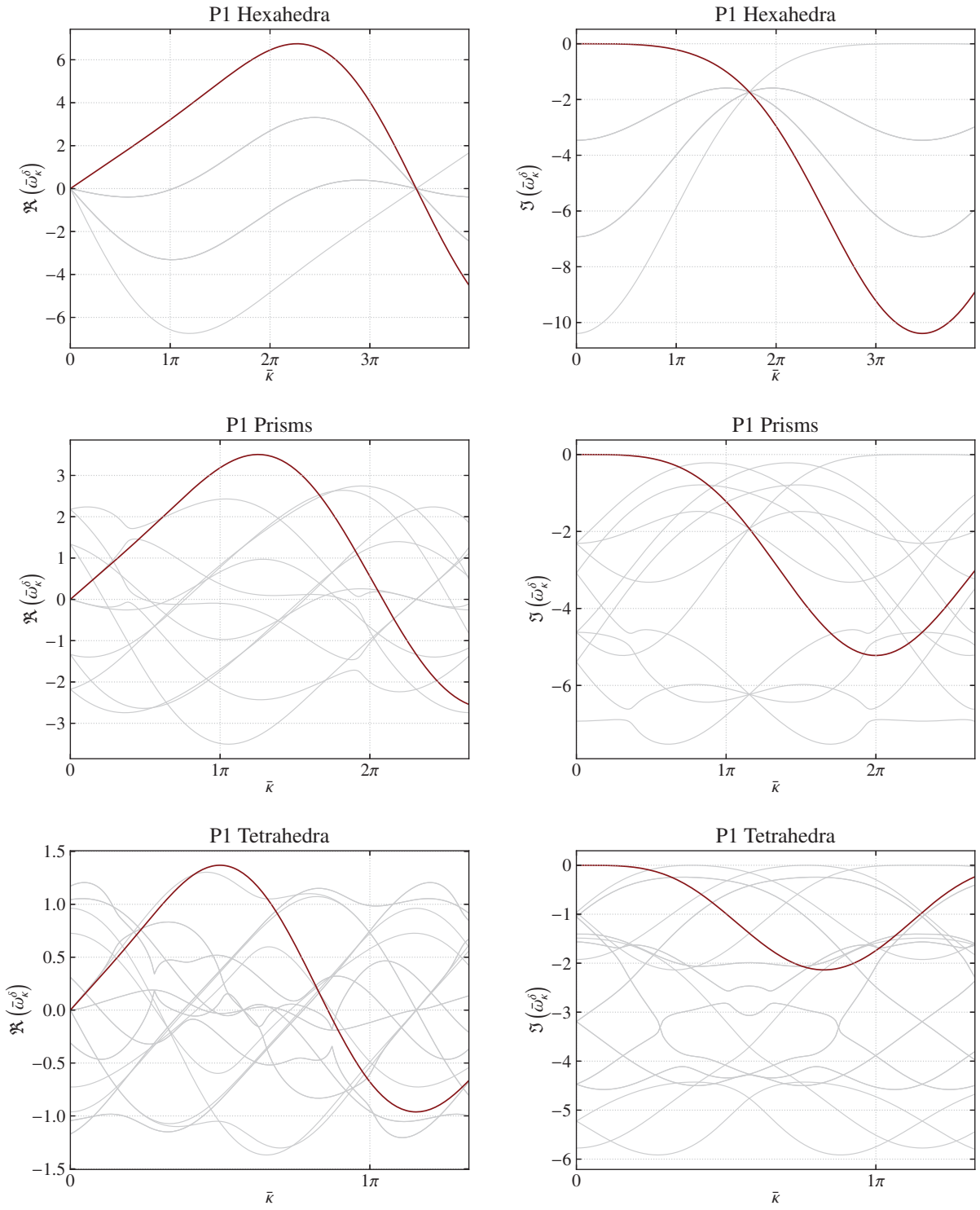


Figure 5.8. Eigenmodes for a wave aligned with the main diagonal of the cuboid at $\theta_1 \approx 0.615, \theta_2 = \frac{\pi}{4}$ on three-dimensional $P1$ schemes.

with non-DG correction functions [18, 52]. This may be a consequence of the agglomeration of N_{sub} elements within the cuboid and may introduce increased error to the solution [53]. The eigencurves for three-dimensional elements and diagonal advection are generally periodic in κ with the period given by

$$\pi(P+1)\min\left(\frac{1}{\cos(\beta_1)\cos(\beta_2)}, \frac{1}{\cos(\beta_1)\sin(\beta_2)}, \frac{1}{\sin(\beta_1)}\right). \tag{5.26}$$

All eigeninformation is generally contained within this range. Nevertheless, when the wave is not aligned with the diagonal or orthogonal directions, the behaviour can be considered instead *quasi-periodic*.

Figure 5.9 shows the dissipation and dispersion curves for an arbitrarily chosen direction $\theta_1 = \theta_2 = \frac{\pi}{9}$. The curves at high-wavenumber regions cannot be considered exact replicas of the modes with physical meaning at lower wavenumbers. Hence, as the wavenumber increases, the aliasing is not periodic. In fact, a wider range of κ would display comparable but not equal dispersion relations. We note that not all eigencurves cross the zero dispersion axis. In some cases, these curves reach a minimum point and the dispersion relation stays in the positive axis. Hence, for the vast majority of advection directions, the dispersion and dissipation properties of each scheme differ significantly from the 1D approach.

5.2.3 Comparison of Three-Dimensional Spatial Discretizations

The dominant dissipation and dispersion curves for all considered element types from second to fourth-order accurate spatial discretizations are shown in Figure 5.10. The wavenumber and the numerical frequency $\bar{\omega}_\kappa^\delta$ are normalized by $\gamma_{c,e}$. For simplicity, we consider only waves aligned with the advection direction for all angles $\theta_1 = \beta_1$, $\theta_2 = \beta_2$. As a consensus, given the quasi-periodicity and non-smooth behaviour of certain eigencurves, they are displayed until the numerical frequency vanishes or the curve has reached its minimum point, and only the positive dispersion relations are shown.

Increasing the polynomial degree results in steeper onset of numerical error, but a larger resolved wavenumber range for all element types. As we sweep from angles $\theta_1 = \beta_1$, $\theta_2 = \beta_2 \in [0, \pi]$, hexahedral elements display curves with smooth shapes, similar to the quasi-one-dimensional

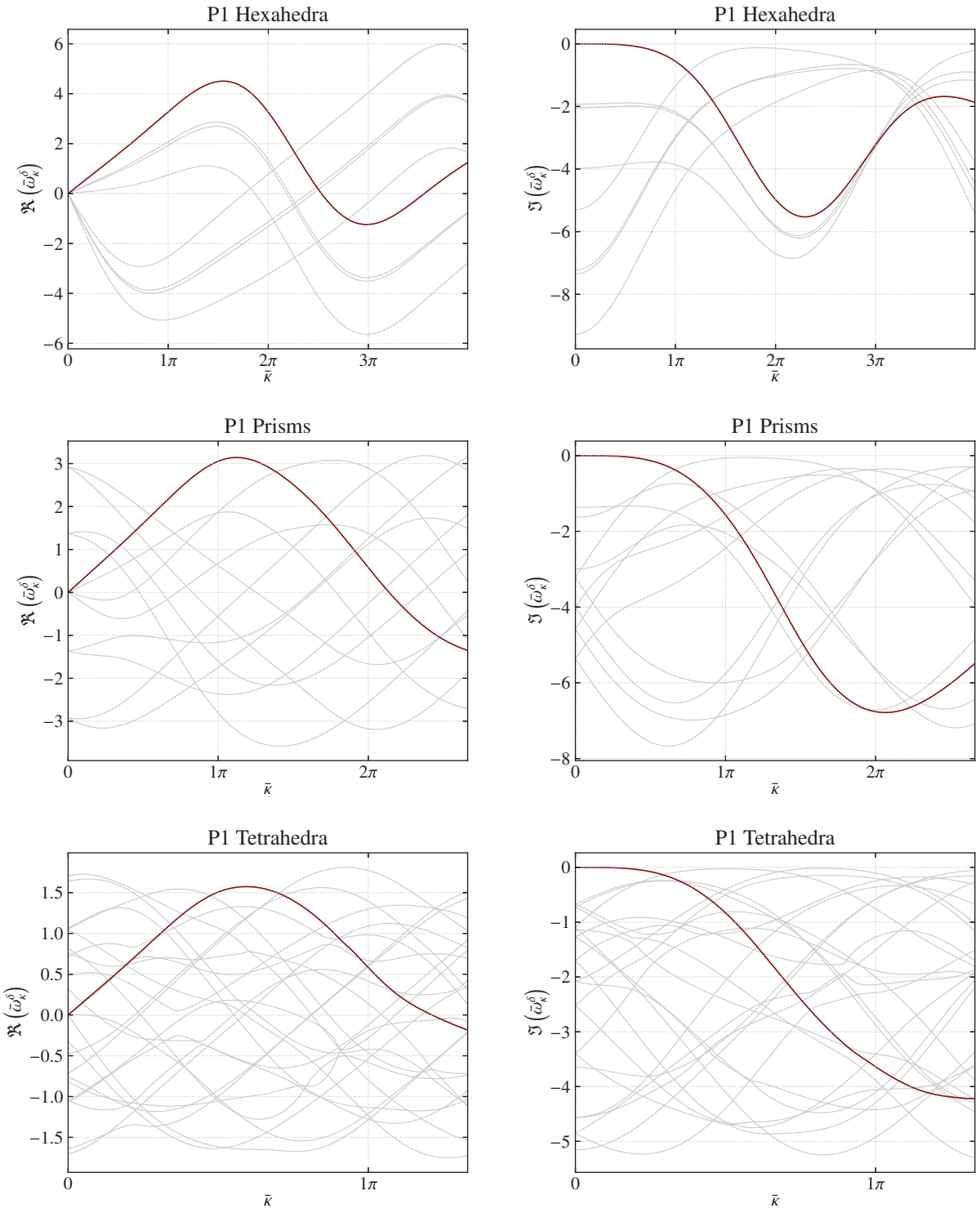


Figure 5.9. Eigenmodes for a wave with orientation $\theta_1 = \frac{\pi}{9},\ \theta_2 = \frac{\pi}{9}$ using three-dimensional $P1$ schemes.

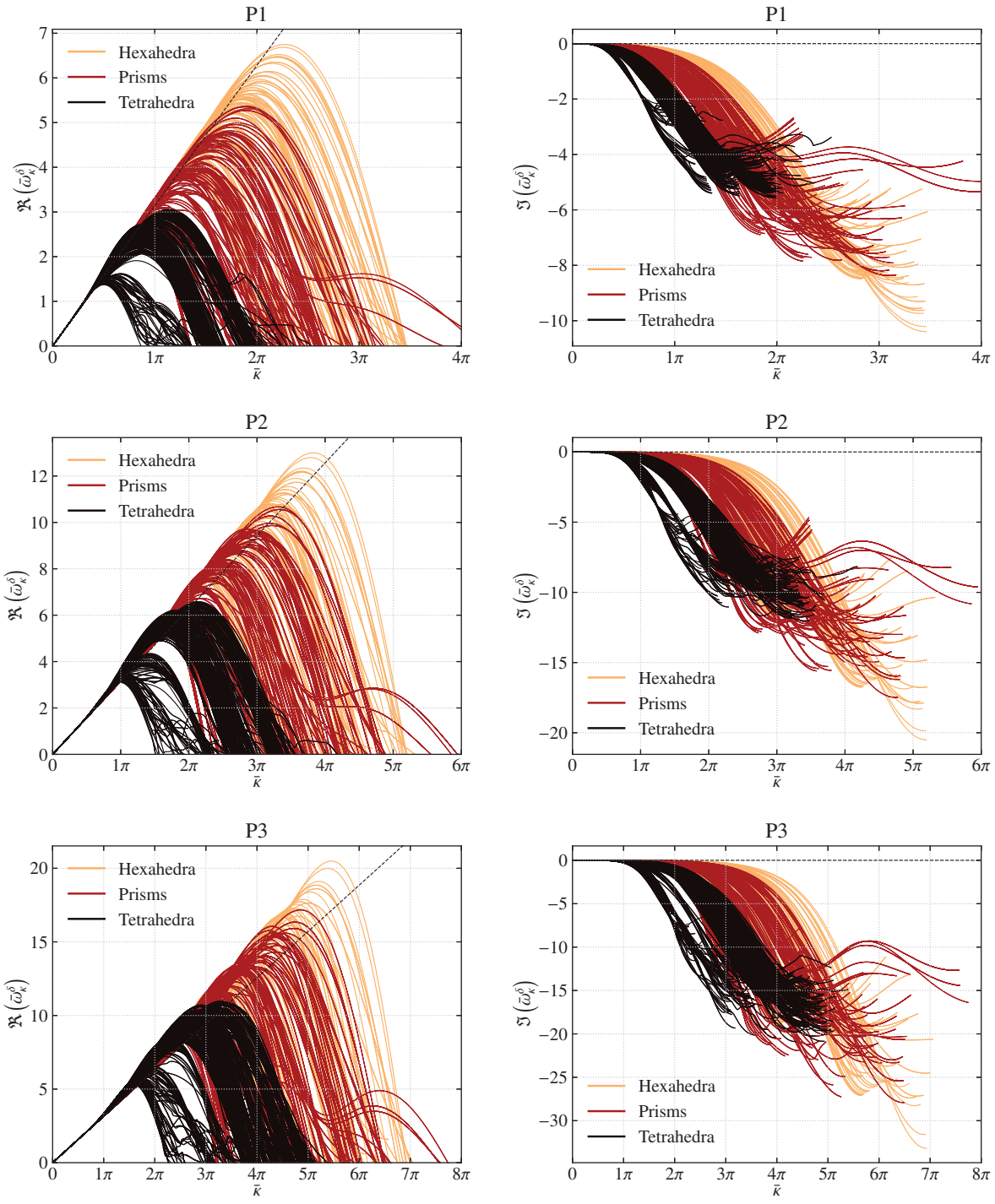


Figure 5.10. Dissipation and dispersion curves for hexahedra, prisms and tetrahedra.

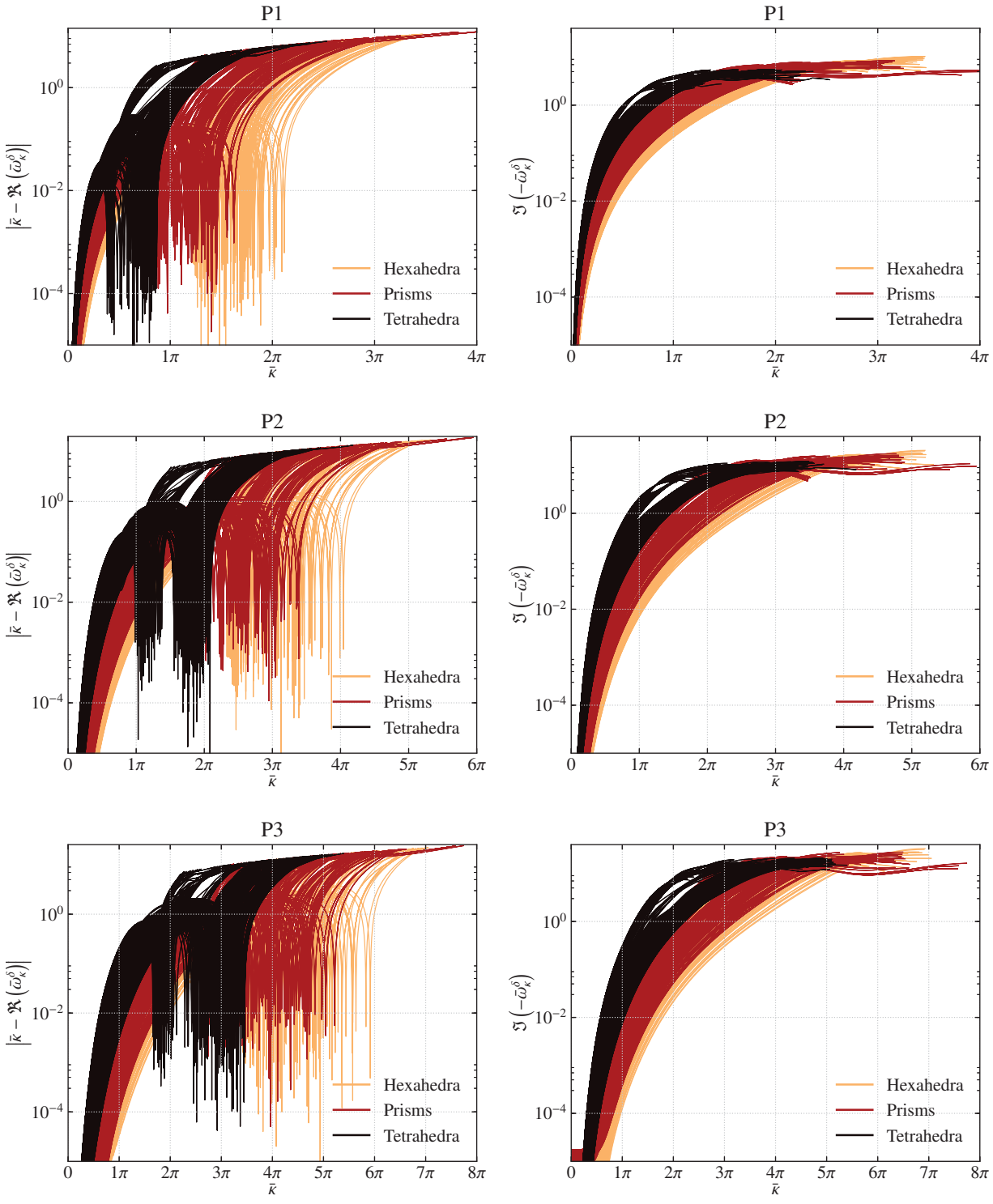


Figure 5.11. Semilogarithmic plot of the dispersion error $\left|\bar{\kappa} - \Re(\bar{\omega}_\kappa^\delta)\right|$ and dissipation error $\Im(-\bar{\omega}_\kappa^\delta)$.

configuration in Figure 5.7 with some growth or shrinkage as the effect of the angle variation appears. For the other two element types, discontinuities and non-smooth modes appear, most notably for the tetrahedra.

We can take a closer look at the region of low wavenumbers with the semilogarithmic plots in Figure 5.11. The dispersion error $\left|\bar{\kappa} - \bar{\omega}_{\kappa}^{\delta}\right|$ is shown in the left side and the dissipation error $-\Im\left(\omega^{\delta}\right)$ in the right side. Note the inversion of the vertical axis of this plot. Hexahedral elements display the minimum dispersion and dissipation error amongst all element types for the majority of wavenumbers. Some eigencurves of the prismatic element overlap the results of the hexahedra. Tetrahedral elements display the most dissipation on a per degree of freedom basis for most wavenumbers. The same outcome is seen for the dispersion plots. For ILES, we consider the best performing element to have the smallest range of numerical error, given the chaotic three-dimensional behaviour of the eddies in turbulent flow. Hence, in terms of DOF cost, hexahedral elements display the least error in both dissipation and dispersion curves, followed by the prisms and finally the tetrahedra. In addition, from Figure 5.10, we see that the performance of the elements is direction-dependent.

The effect of the direction on the dispersion and dissipation characteristics of the scheme is shown in Figure 5.12 for a given wavenumber $\bar{\kappa} = \pi$. Similar behaviour was found for other wavenumbers and the dependence increases with κ, but are omitted for brevity. Hexahedral elements perform best at diagonal angles, which agrees well with previous studies on quadrilateral elements (for $\theta_2 = 0$) [23]. They are most dissipative and dispersive when the wave is aligned with the grid. Prismatic elements, on the other hand, benefit from waves normal to the diagonal of the triangular faces. They are most dissipative parallel to these diagonals. Finally, similar to the prisms, tetrahedral elements introduce the least amount of numerical error along the faces of the cuboid with wavevectors normal to the diagonal interfaces, and the most when the wavevector points parallel to the main diagonal of the cuboid $\left(\theta_1 = \theta_2 = \frac{\pi}{4}\right)$. Despite the dependence of the results on the direction, the dissipation error of the tetrahedral elements is higher than that of the prisms and hexahedra.

Hence, our spectral analysis suggests that the dispersive and dissipative properties of three-dimensional discretizations using the FR approach are highly dependent on the element type and advection direction. Furthermore, we generally observe that the hexahedra perform best, followed by prisms and finally tetrahedra on a per degree of freedom basis.

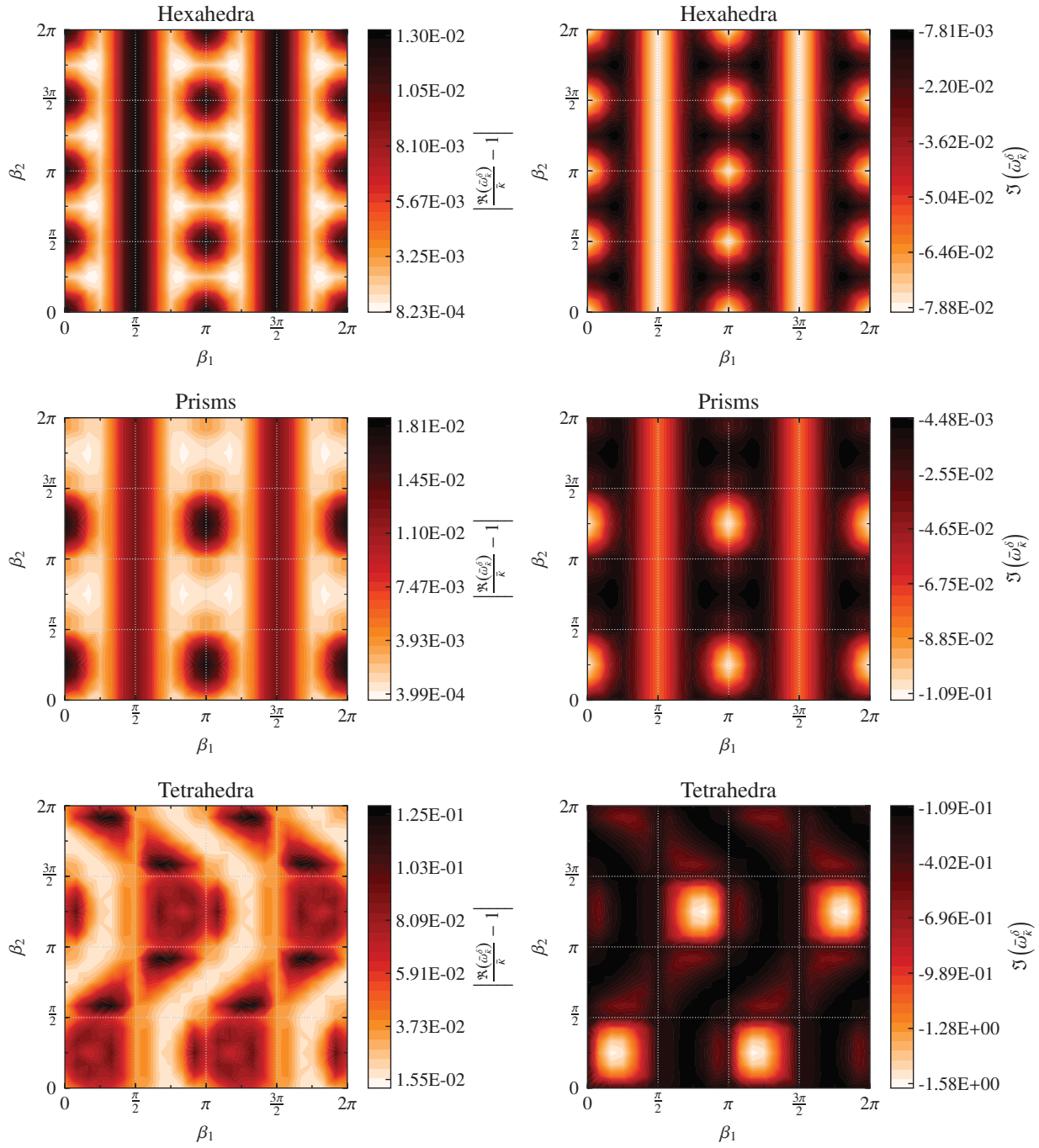


Figure 5.12. Dissipation rate and dispersion contours for $\bar{\kappa} = \pi$ on $P2$ three-dimensional schemes.

5.2.4 Eigenspectra

The eigenspectra or Fourier footprint [54] of the semidiscrete matrix can be determined by computing its eigenvalues. This region contains important information. In order for a scheme to be stable, all eigenvalues λ_j must lie on the negative real axis to ensure that none of the Fourier modes will grow in time. In addition, the size of the eigenspectra increases at a faster rate than the polynomial degree, as explained for the one-dimensional case in Section 5.1.2.

Figures 5.13 to 5.17 display the collection of eigenvalues $\boldsymbol{\lambda}$ for $P1$ to $P3$ schemes on quadrilaterals, triangles, hexahedra, prisms and tetrahedra. In two dimensions, quadrilateral elements display eigenspectra that are more compact than those of the triangles by a factor of approximately 1.4 for the lower order schemes. In three dimensions, hexahedral elements yield more compact eigenspectra compared to prisms and tetrahedra. This suggests the hexahedral cuboid with N_{DOF} is able to be stable with larger time-step sizes than the other two element types. Out of the three element types, it appears that tetrahedral elements have the largest region of eigenvalues. We note that the size of the time-step is defined by the number of degrees of freedom as well and that N_{DOF} for the tetrahedra and prisms is higher than that of the hexahedral cuboid, so these properties may be different for equally dense computational grids.

5.2.5 Orders of Accuracy

Discontinuous Galerkin methods have known to be super accurate for hyperbolic problems [51]. This refers to orders of accuracy higher than $P+1$, where P is the polynomial degree that interpolates the solution. This behaviour is not well-understood [9] and has shown to converge at rates of $2P+1$ for this type of problems. From the spectral analysis, the local order of accuracy can be found by inspecting the numerical error introduced in the solution within the moderately resolved range of Fourier modes. For a given wavenumber, the error can be determined by [20]

$$E_k = \left| \kappa - w^{\delta *}(\kappa) \right|, \tag{5.27}$$

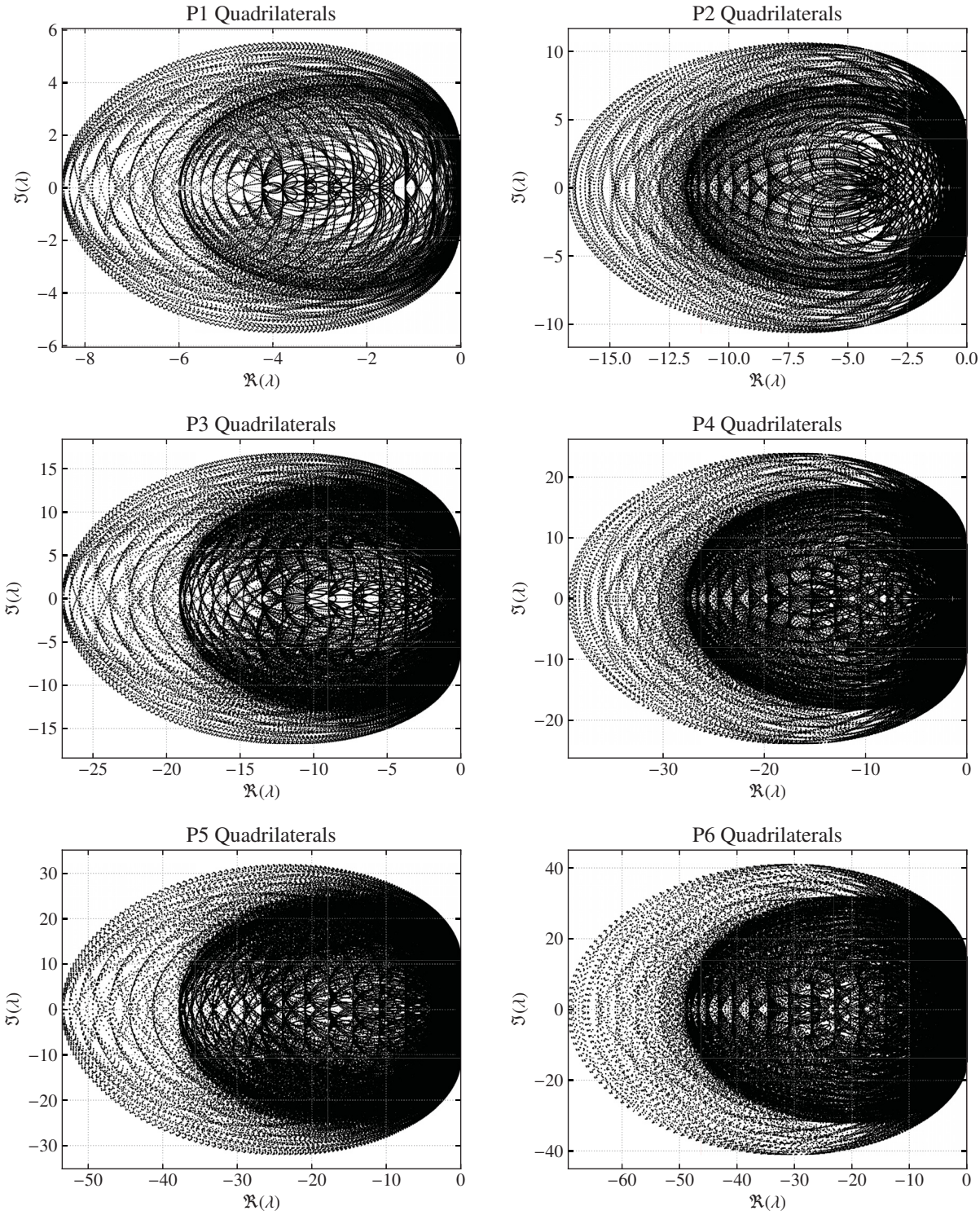


Figure 5.13. Eigenspectra of quadrilateral semidiscretizations.

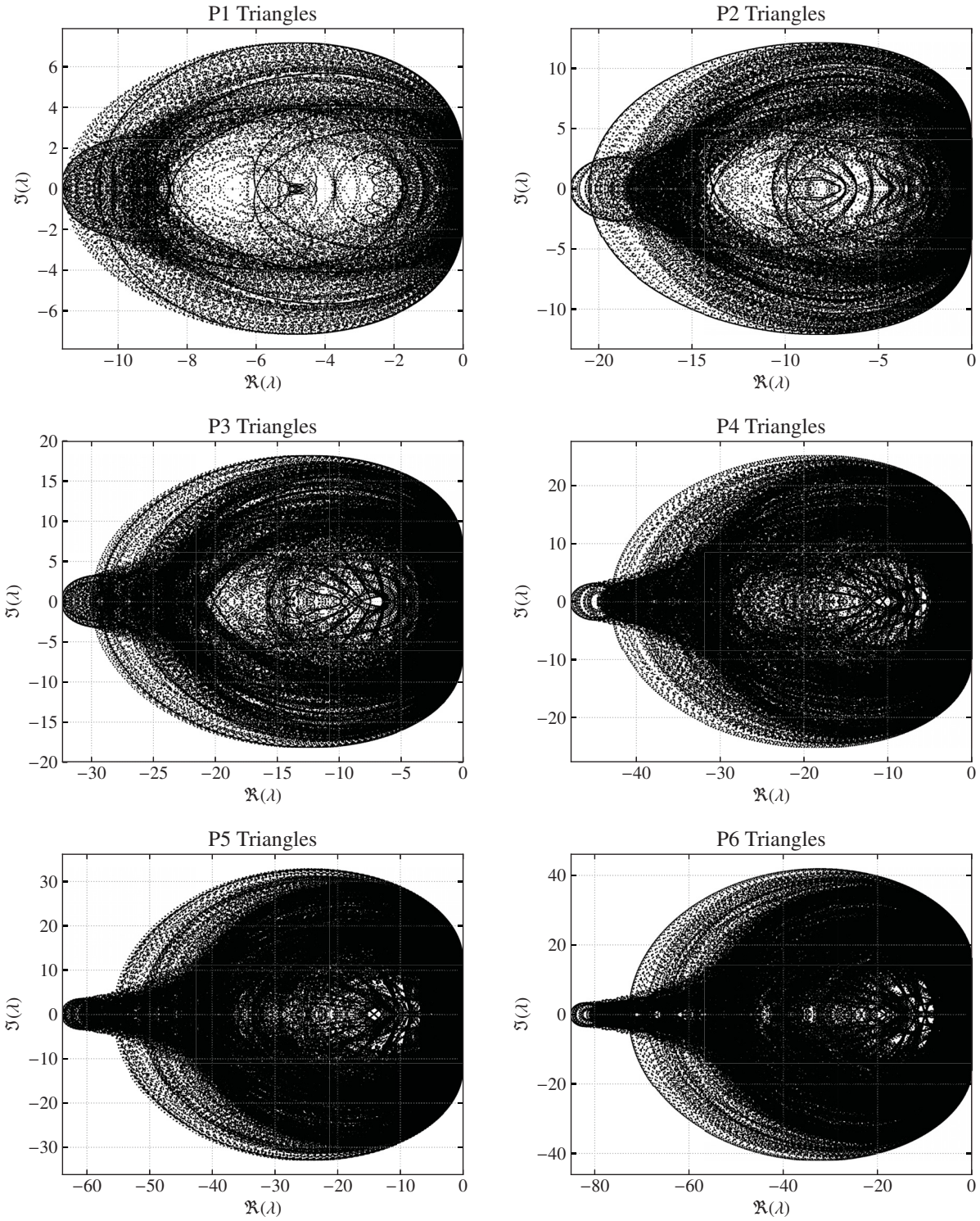


Figure 5.14. Eigenspectra of triangular semidiscretizations.

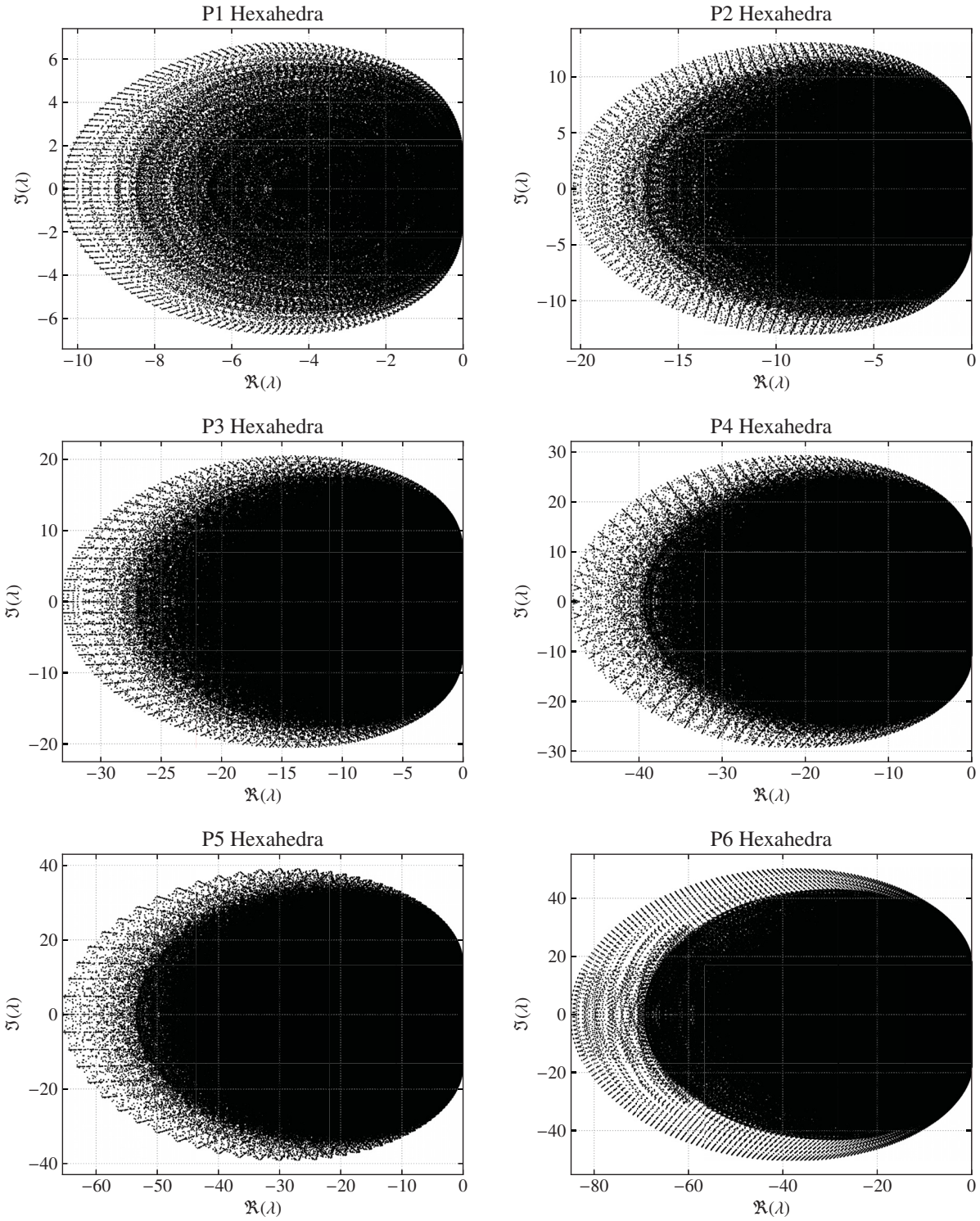


Figure 5.15. Eigenspectra of hexahedral semidiscretizations.

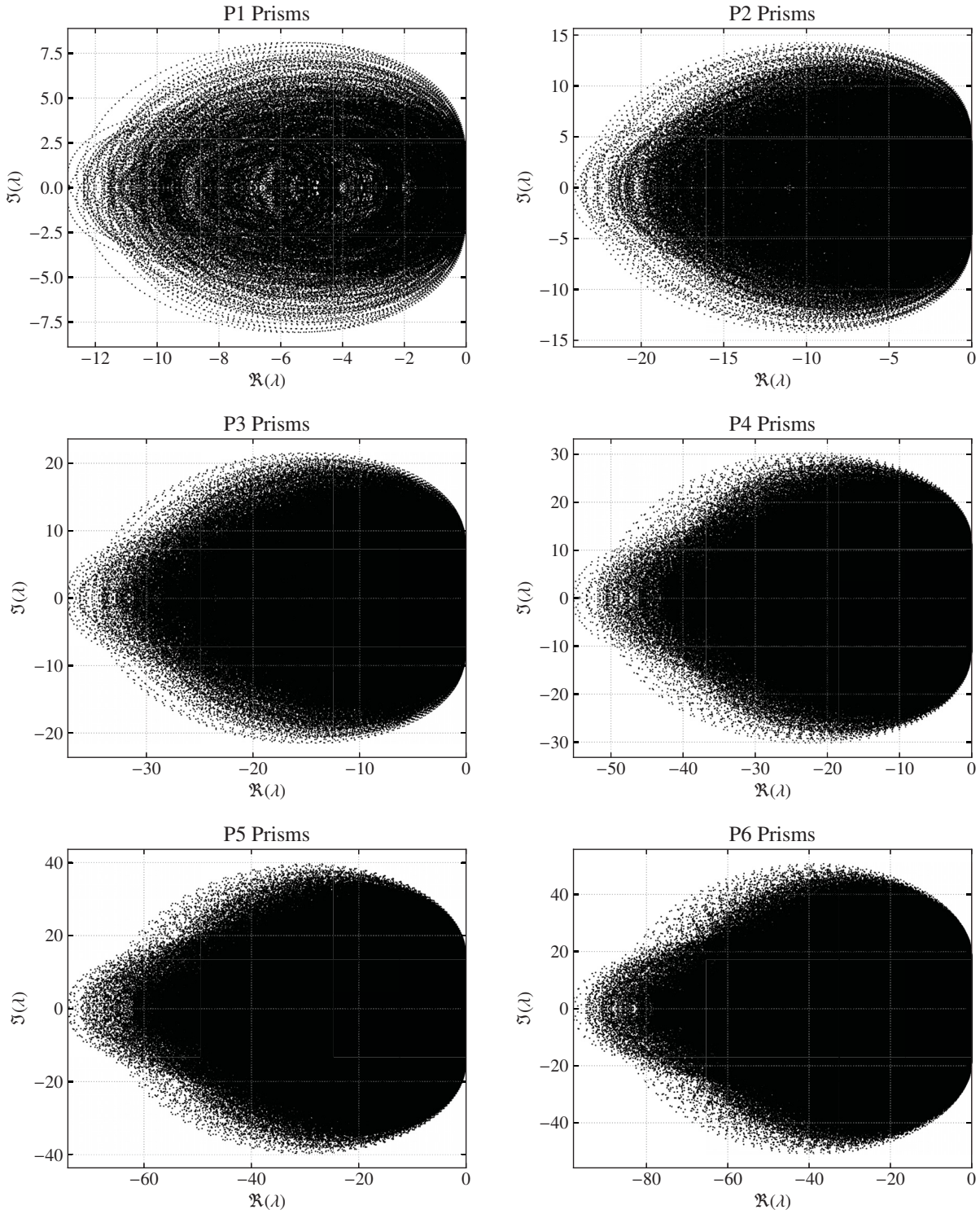


Figure 5.16. Eigenspectra of prismatic semidiscretizations.

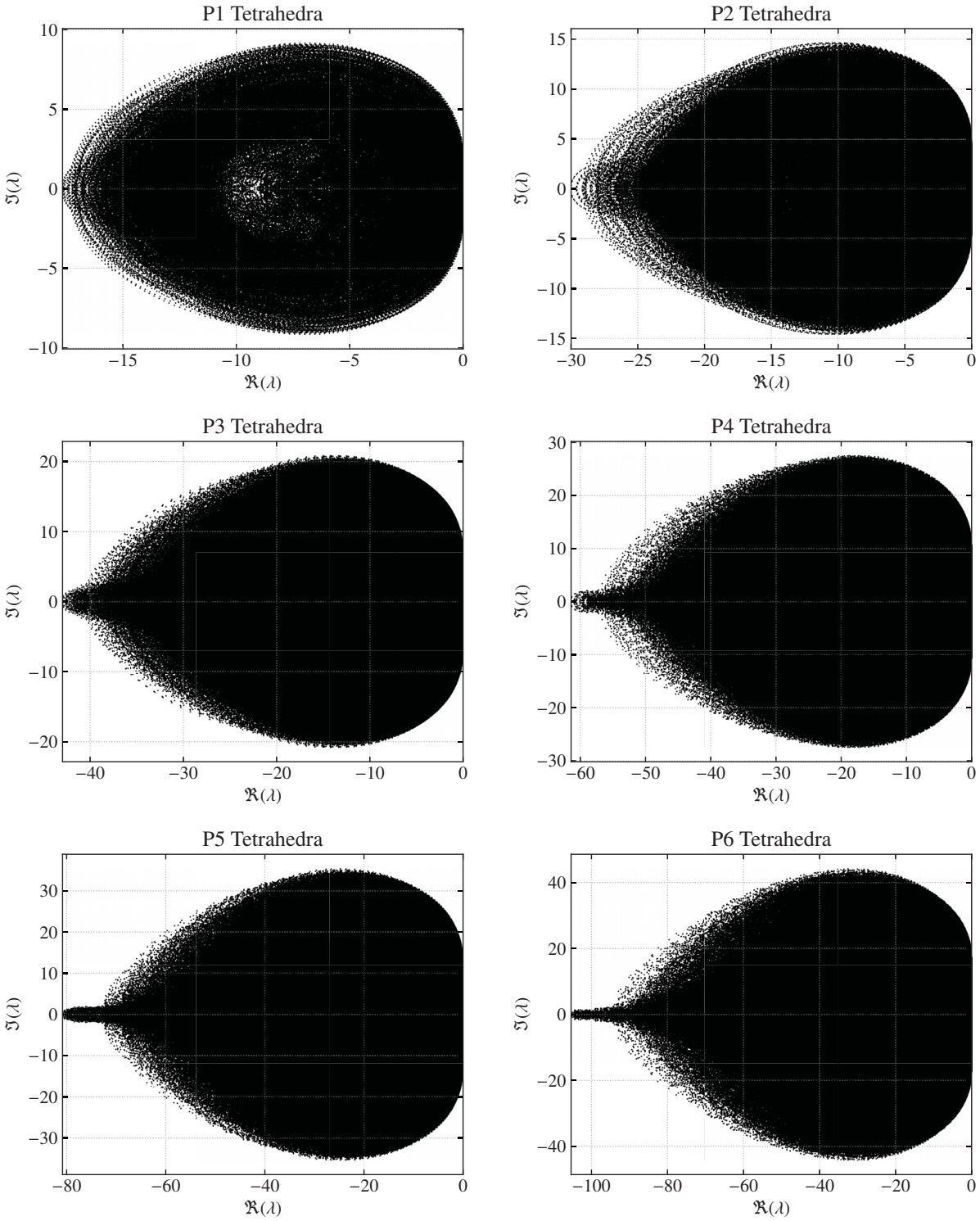


Figure 5.17. Eigenspectra of tetrahedral semidiscretizations.

where κ is the wavenumber and $\omega^{\delta *}$ is the dominant numerical frequency. The local and global orders of accuracy can be respectively found by [20]

$$OOA_l = \frac{\log\left(E_\kappa\right) - \log\left(E_{\kappa/2}\right)}{\log(2)}, \tag{5.28}$$

$$OOA = OOA_l - 1. \tag{5.29}$$

In Figure 5.18, we observe the local order of accuracy for P1-P4 schemes for all element types, which is consistent with the expected super convergence rates of $2P+1$ We recall from the spectral analysis that the cuboids composed of prisms and tetrahedral elements contain more degrees of freedom than those in a hexahedron. Despite this advantage, prismatic and tetrahedral elements show regions where the error is larger compared to the hexahedral cuboids. In addition, hexahedral elements maintain the slope for a larger range of wavenumbers compared to the other two element types for all polynomial degrees, suggesting that the super-accuracy property holds for a higher range of wavenumbers.

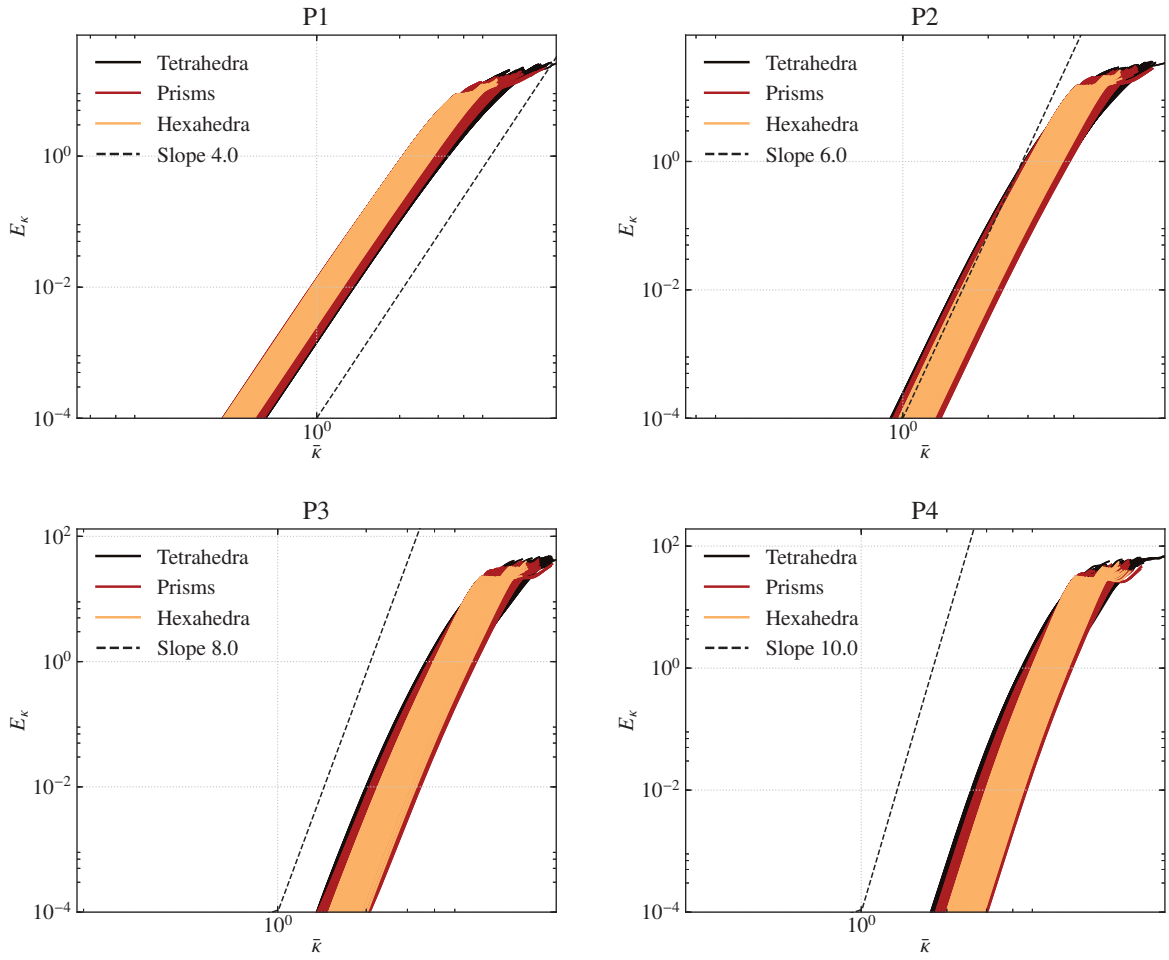


Figure 5.18. Local order of accuracy obtained from spectral analysis for three-dimensional semidiscretizations.

5.3 Characteristic Functions of the Spatial Discretization

The semidiscrete form of the linear advection equation for the dispersion and dissipation analysis is given by

$$i\frac{\nu h}{\|\boldsymbol{\alpha}\|}\boldsymbol{U} = M\boldsymbol{U}, \tag{5.30}$$

where $\boldsymbol{U}$ is the vector that contains the amplitudes of the discretized wave solution, ν is the exact numerical frequency, $\boldsymbol{\alpha}$ is the advection velocity, h is the mesh spacing and M is the semidiscrete matrix. The numerical solution can be represented as a linear combination of the eigenvalues and eigenvectors of M. Since M has rank $m = N_{DOF}N_{sub}$, m linearly independent eigenvectors $\boldsymbol{U}_j$ with

their corresponding eigenvalue λ_j reconstruct a basis polynomial space of degree P and a matrix $\hat{U}$. These polynomials represent characteristic functions (or eigenfunctions) of the semidiscrete operator. By constructing them, each solution mode may be analyzed. Hence, the identification of the physical modes becomes possible by looking at their eigenshapes.

Following the procedure in [16] in one dimension, we present a general form for any dimensional space on lines, quadrilaterals and hexahedra. The eigenvectors of M form a complete basis. Hence, the eigensolution, i.e. the numerical solution constructed from the eigenvalues and eigenvectors of M, may be computed from

$$u_e^{\delta}\left(\boldsymbol{\xi}, t\right) = \sum_{j=1}^{N_{DOF}} u_{e,j}\psi_j\left(\boldsymbol{\xi}\right), \tag{5.31}$$

where $\psi_i\left(\boldsymbol{\xi}\right)$ is the corresponding eigenfunction and $u_{e,j}$ is the numerical solution value recovered from the j-th numerical frequency $\omega_j^{\delta} = \frac{\alpha\lambda_j}{ih}$ with its corresponding wavenumber κ, calculated for an element Ω_e centered at $\boldsymbol{x}_e$

$$u_{e,j} = u_j^0 e^{i(\boldsymbol{\kappa}\boldsymbol{x}_e - \omega_j^{\delta} t)}, \tag{5.32}$$

and u_j^0 represents the initial amplitude of the prescribed wave after its projection onto the discrete space [16], which can be obtained

$$\boldsymbol{u}^0 = \hat{U}^{-1}\boldsymbol{\sigma}, \qquad \sigma_j = \int_{\Omega_e} e^{i\frac{\boldsymbol{\kappa}h\boldsymbol{\xi}}{2}} \tilde{\ell}_j\left(\boldsymbol{\xi}\right) d\boldsymbol{\xi}. \tag{5.33}$$

Finally, the eigenfunctions can be computed from

$$\psi_j\left(\xi\right) = \sum_{n=1}^{m} \hat{U}_{n,j}\tilde{\ell}_n(\xi), \tag{5.34}$$

and the solution may be obtained using Equation 5.31.

Since the one-dimensional results are easier to visualize, we introduce the eigenfunctions for different P-schemes, similar to the results presented by [55], before we show the results for two-dimensional cases. In Figure 5.19, we observe the numerical solution reconstructed from the procedure above. From previous sections, we have established that for 1D cases, the cutoff wavenumber is defined to be $(P+1)\pi$, which is, in fact, an overestimation when coupled with ILES, since the numerical error will have dominated the solution before this point. In this sense, we still chose

a relatively high wavenumber for a $P1$ scheme to show its improvement with the increase of the order of accuracy of the spatial discretization. Results are shown at $t^* = 0$ and after the wave was convected for one cycle, i.e. $t^* = 1$. It is clear that the resulting data is smoother for higher-order schemes, and that despite knowing that a $P1$ scheme has a cutoff wavenumber of 2π, even at $\pi/2$ the results are not very good. The innate dissipative behaviour of a $P1$ scheme along with an upwind Riemann solver can be catastrophic to represent wave-like solutions. Both numerical dispersion and dissipation are observed for the $P2$ schemes, and the jumps continue to close as P increases. These sinusoidal plots are the result of a combination of modes, shown in Figure 5.20.

Again, these plots in Figure 5.20 are shown for both initial projection $t^* = 0$ and after an advective cycle. The dominant modes (those with shapes closest to the expected numerical result) can be seen to stand out. As time passes, we observe that for the low order schemes, the smoothness of the solution is affected, and dispersion, as well as dissipation error, begin to appear. At the same time, the modes with ranges close to zero completely disappear. Hence, the secondary modes are in fact a contribution to the numerical solution, as shown for $t = 0^*$. The fact that the solution at $t^* = 1$ contains more error shows that these modes cannot be regarded as spurious.

Figure 5.21 shows the resulting numerical solutions for quadrilateral elements at $t^* = 0$ and after $10t^*$, meaning that the solution cycled over the domain 10 times. The jumps in the solution are visible for the $P1$ scheme, and the numerical dissipation appears to dominate at the later time. Similar to the one-dimensional case, the solution contains more numerical error for the low order schemes. In fact, from these results, we observe the solution to be quite smooth for $P2$-$P4$ schemes, for both initial and final times, with more numerical accuracy for the higher order schemes.

Looking at multidimensional eigenmodes may be more difficult if the purpose is to identify the dominant mode. However, we can readily note that some of these modes exist in ranges that are very close to zero, marking them as candidates for secondary modes. In addition, we observe that more than a single mode is dominant since Modes 0 and 8 contain values that are similar to the prescribed wave-like solutions.

We now validate this spectral analysis through some numerical experiments, which we present in the following chapter.

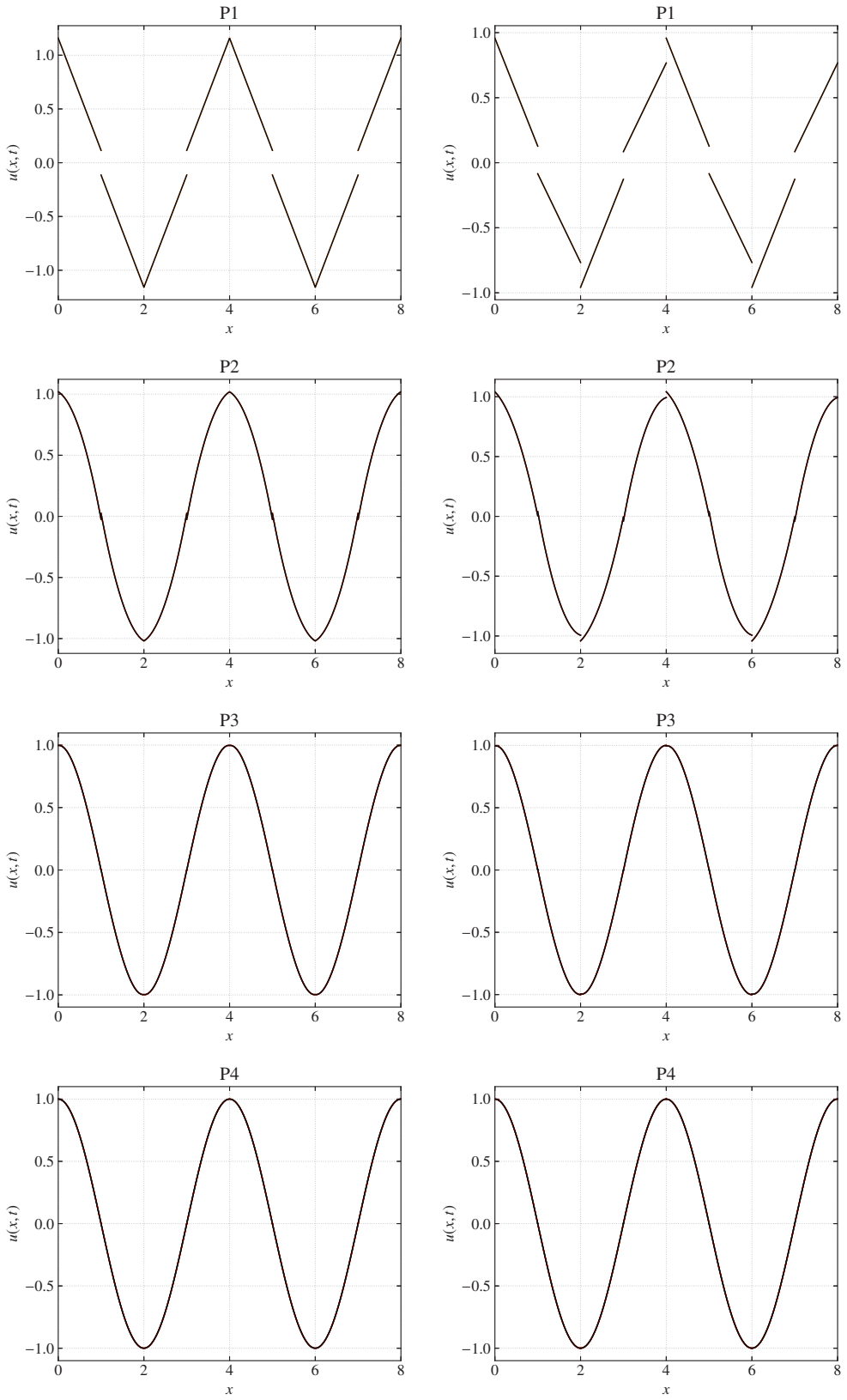


Figure 5.19. Eigensolution for the one-dimensional case at $t^* = 0$ (left) and $t^* = 1$ (right).

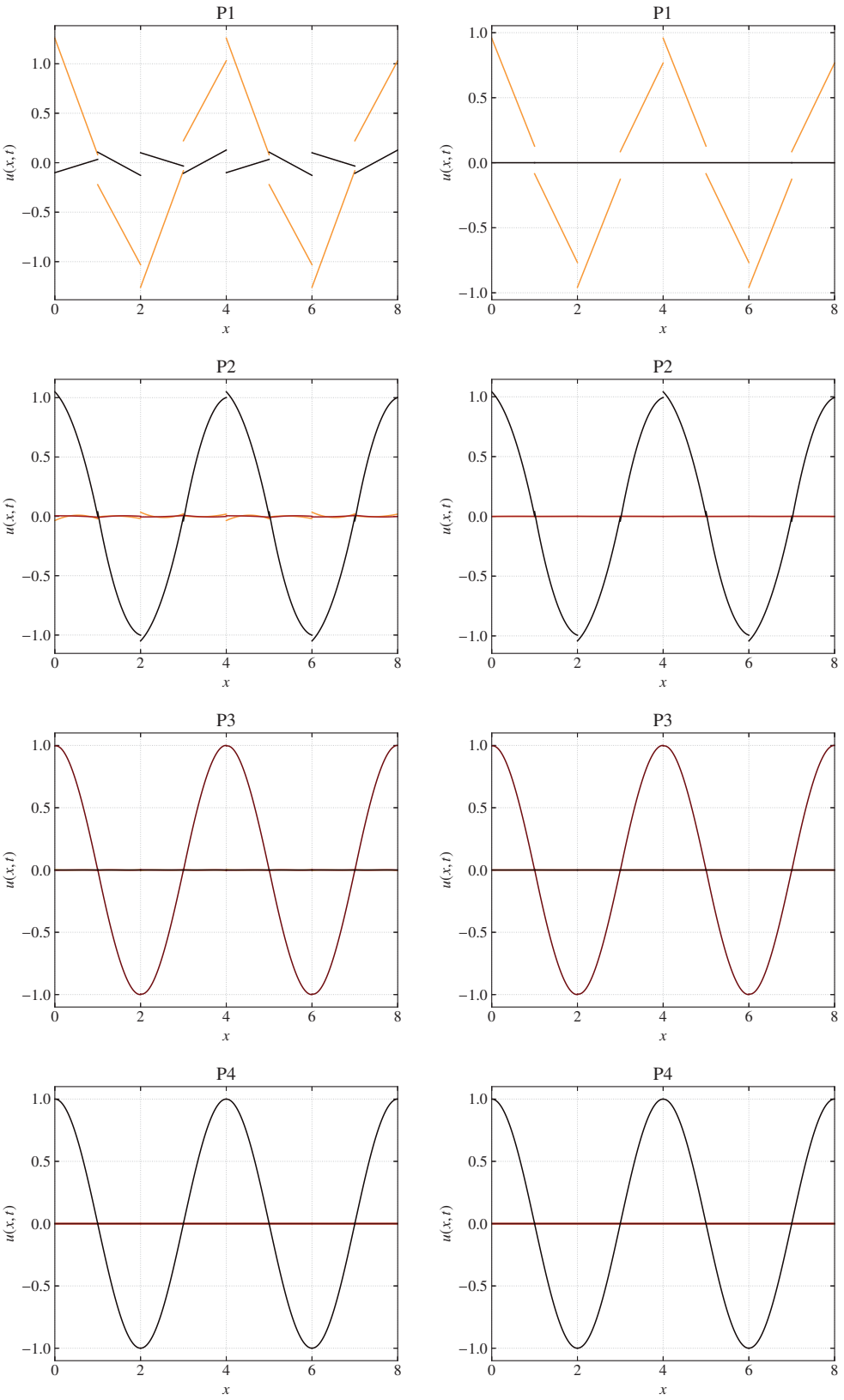


Figure 5.20. Eigensolution modes for the one-dimensional case at $t^* = 0$ (left) and $t^* = 1$ (right).

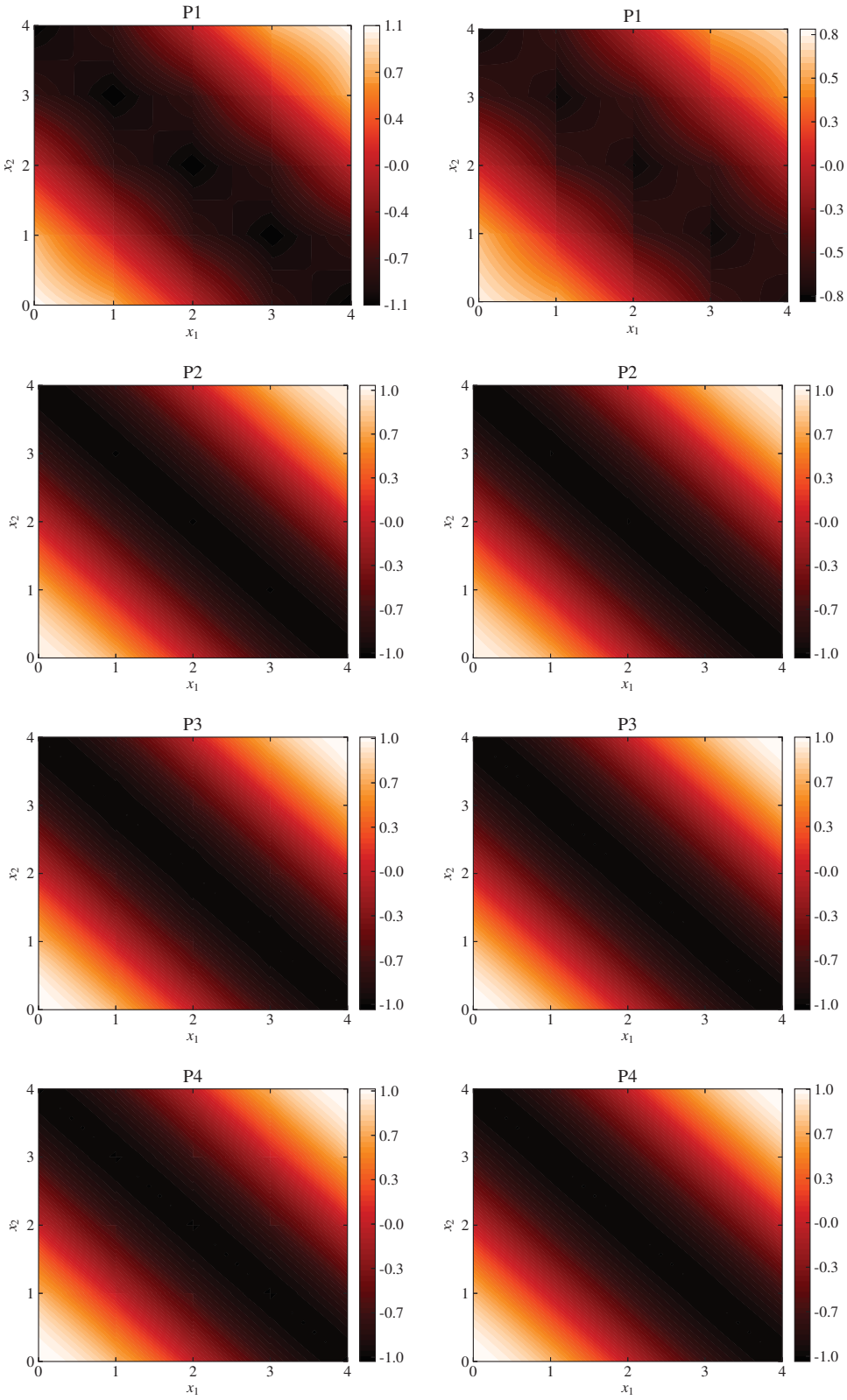


Figure 5.21. Eigensolution on quadrilateral elements at $t^* = 0$ (left) and $t^* = 1$ (right) for a wave oriented at $\theta = \pi/4$.

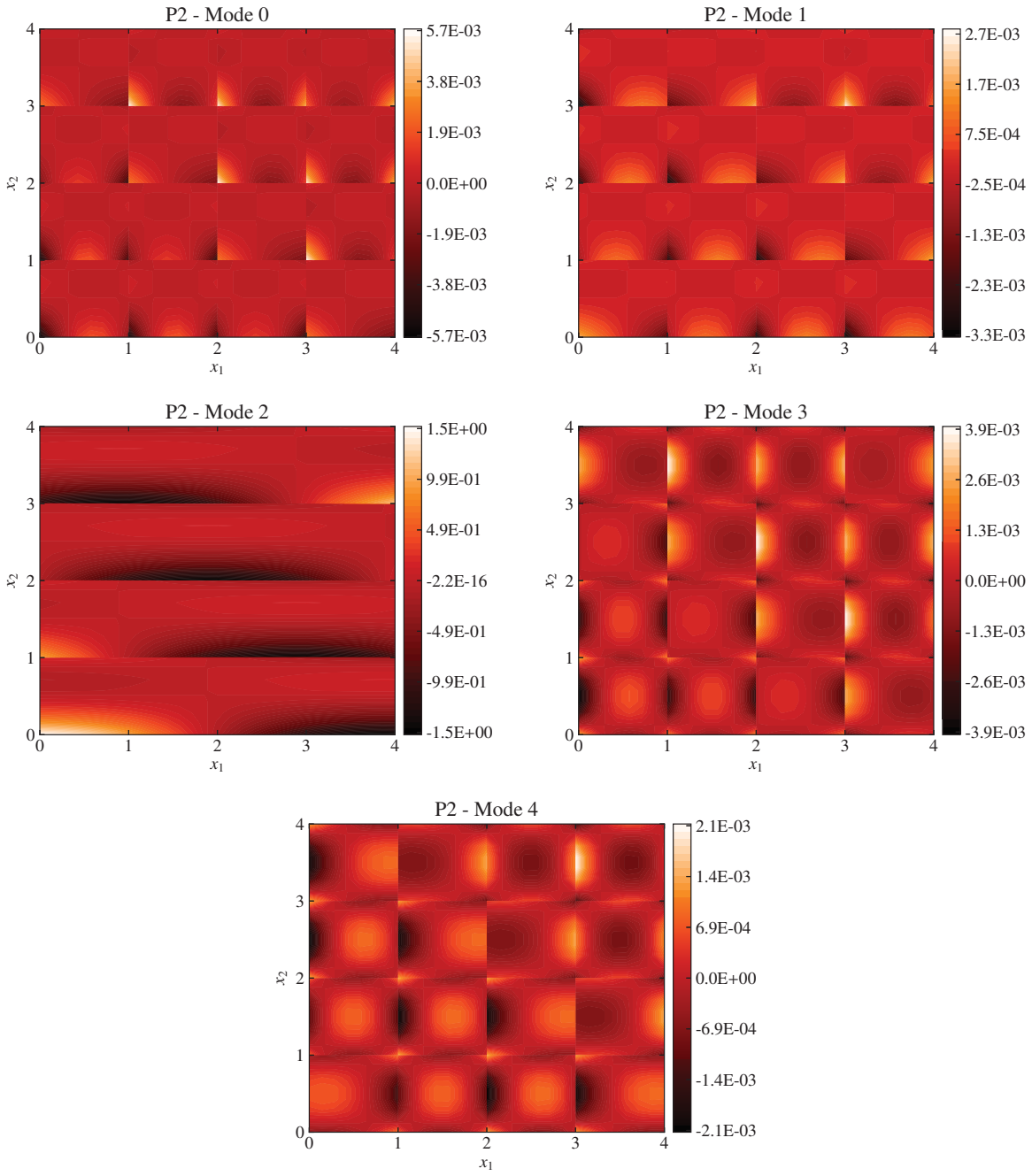


Figure 5.22. Eigensolution modes (0-4) for quadrilateral semidiscretizations at $t^* = 0$ for a wave oriented at $\theta = \pi/4$ on a $P2$ scheme.

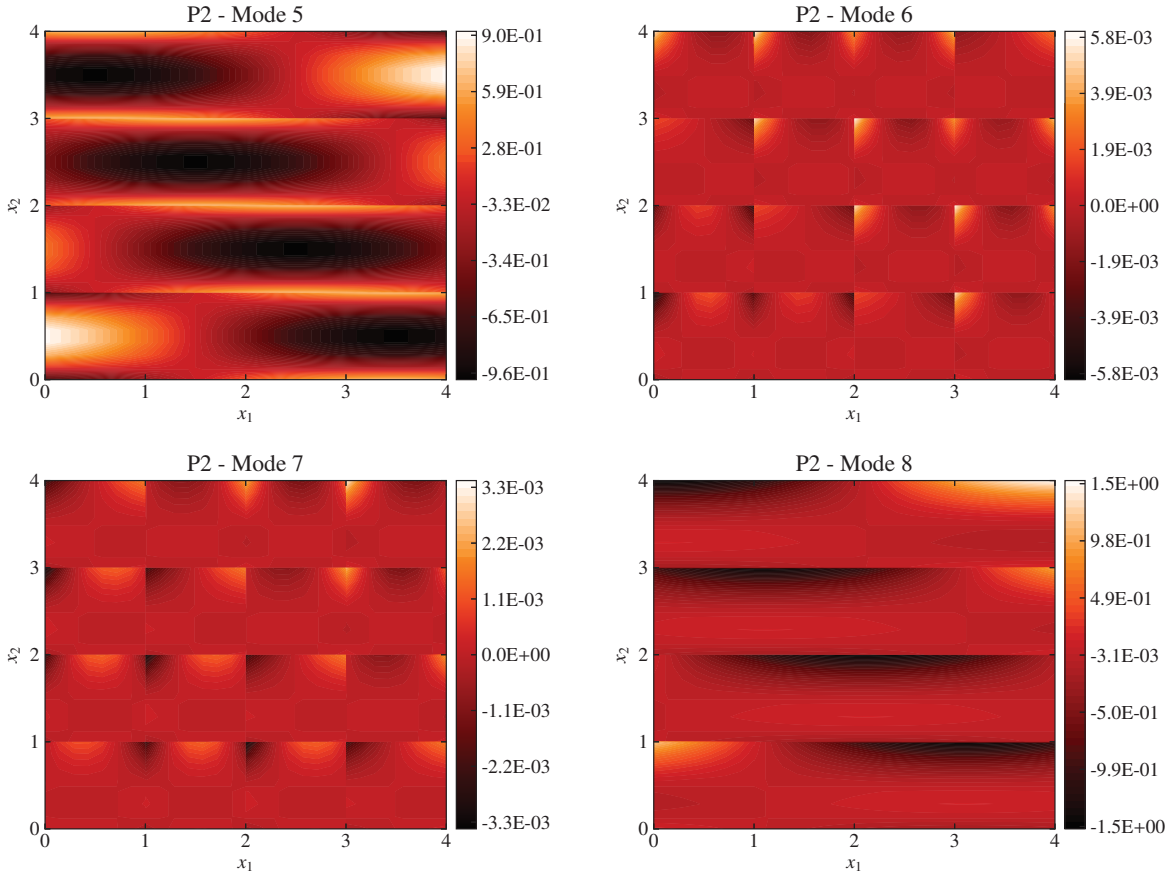


Figure 5.22. Eigensolution modes (5-8) for quadrilateral semidiscretizations at $t^* = 0$ for a wave oriented at $\theta = \pi/4$ on a $P2$ scheme.

Chapter 6

Numerical Experiments

In this chapter, we validate the results obtained in Chapter 5 by analyzing the behaviour of different three-dimensional element types on comparable computational grids. We study three turbulent cases using ILES: the Taylor-Green vortex, turbulent channel flow and turbulent flow over a cylinder.

6.1 Taylor-Green Vortex

6.1.1 Case Description

The Taylor-Green vortex is a standard benchmark case used to test a scheme's ability to resolve turbulent flow. This case has been frequently presented in a series of high-order methods workshops [56]. The flow is initialized with the following conditions

$$v_x = v_0 \sin\left(\frac{x}{L}\right) \cos\left(\frac{y}{L}\right) \cos\left(\frac{z}{L}\right), \tag{6.1}$$

$$v_y = -v_0 \cos\left(\frac{x}{L}\right) \sin\left(\frac{y}{L}\right) \cos\left(\frac{z}{L}\right), \tag{6.2}$$

$$v_z = 0, \tag{6.3}$$

$$p = p_0 + \frac{\rho_0 v_0^2}{16}\left[\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right)\right]\left[\cos\left(\frac{2z}{L}\right) + 2\right]. \tag{6.4}$$

The domain is a periodic cube with dimensions $-L\pi \leq x \leq L\pi$, with Mach number $M_a = 0.1$ and the Prandtl number $Pr = 0.71$. The initial temperature field is specified with a uniform $T = T_0$, and the initial density field is taken as $\rho = PRT_0$. Here, we run the compressible benchmark problem using ILES at Reynold's number $Re = 1600$ for a coarse mesh of 64^3 and a fine mesh of 128^3 with hexahedral, prismatic and tetrahedral element types. All simulations used a Rusanov Riemann solver

for the common inviscid flux, the BR2 scheme for the common viscous fluxes and the same point sets as the turbulent channel case.

The number of elements per direction was computed based on the DOF of the cuboid subdivision. Hence, the number of cuboids per direction was chosen to be the next integer that would yield the closest amount of required solution points in each direction. The ratio of DOF of the resulting computational domain to those in the grid composed of hexahedra are shown in Table 6.1. The total number of points was higher for all elements that used subdivided cuboids (See Chapter 5).

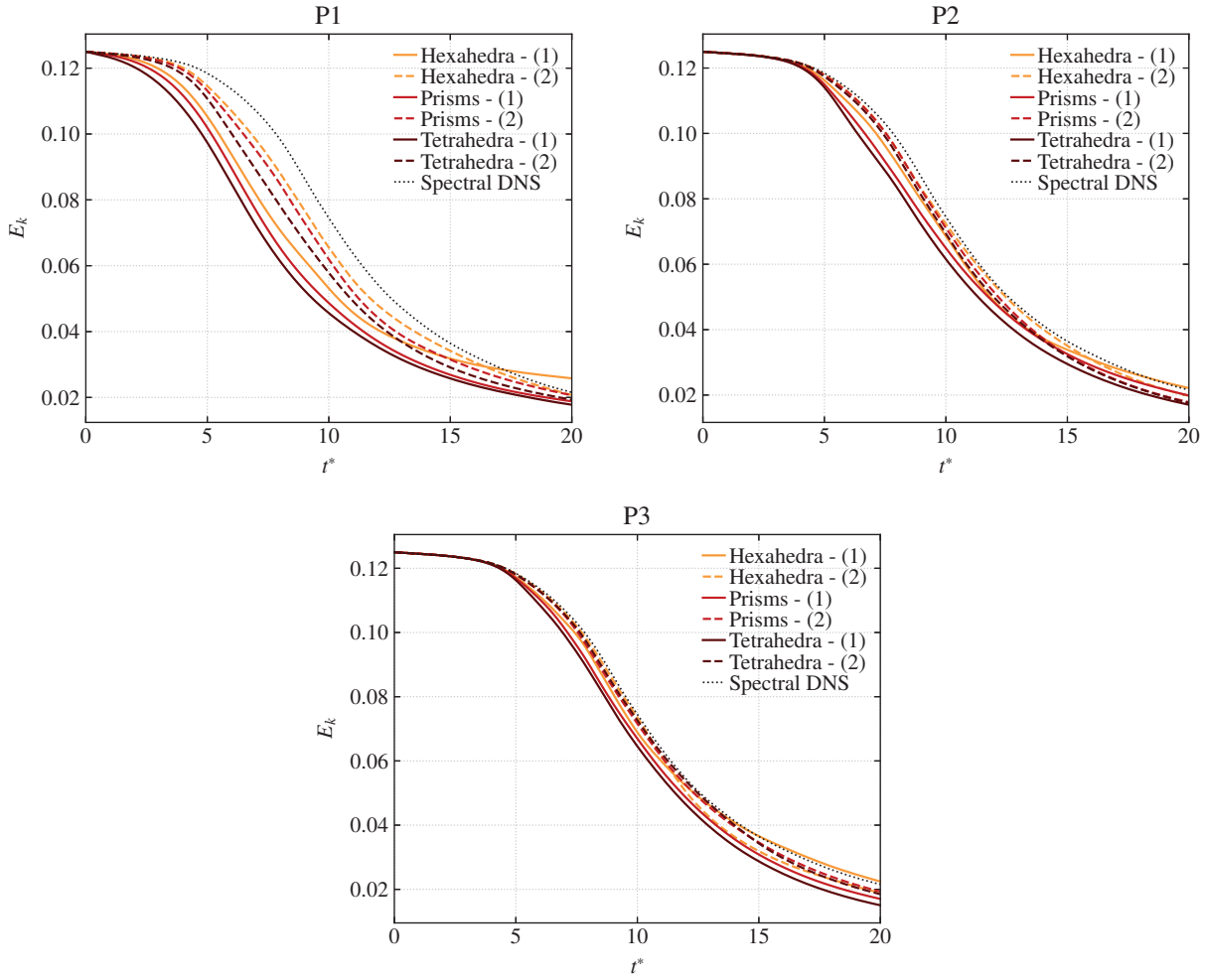


Figure 6.1. Temporal evolution of the kinetic energy E_k integrated over the domain for $P1$-$P3$ schemes for the Taylor-Green vortex. Results are shown for both the coarse case (1) with 64^3 DOF and the finer case (2) with 128^3 DOF.

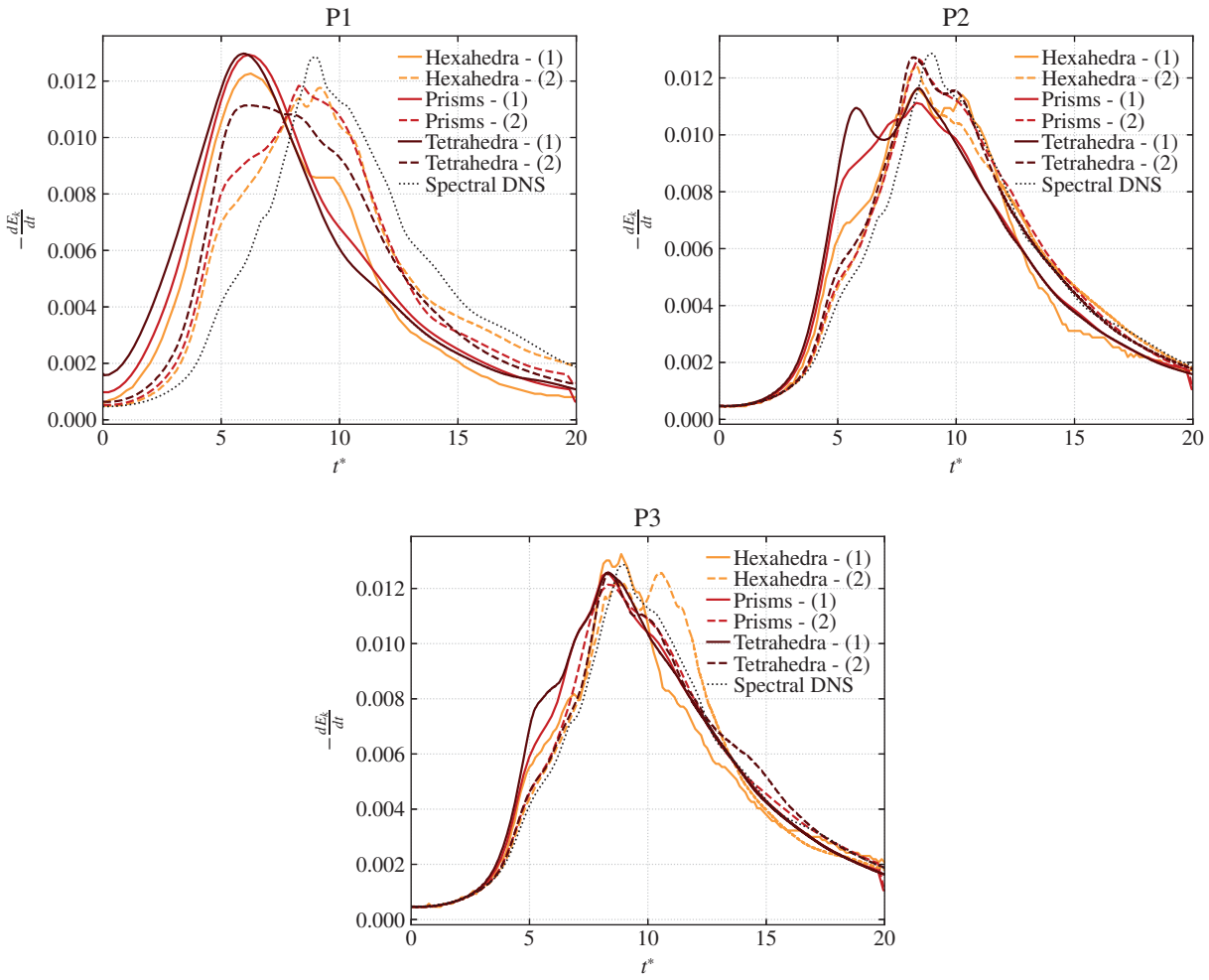


Figure 6.2. Temporal evolution of the kinetic energy integrated over the domain for $P1$-$P3$ schemes for the Taylor-Green vortex. Results are shown for both the coarse case (1) with 64^3 DOF and the finer case (2) with 128^3 DOF.

6.1.2 Results

The evolution of the kinetic energy $E_k(t)$ can be observed in Figure 6.1. It is clear that the amount of energy in time is lower than the DNS results [56], suggesting the presence of numerical dissipation. The results converge to the DNS data as the order of accuracy increases. In particular, the finer mesh agreed well with the spectral results with some slight underprediction. For $P1 - P3$ schemes, the hexahedral elements displayed the maximum amount of energy throughout the evolution of the internal vortical structures compared to the prisms and tetrahedra, which presented more dissipation. This is observed more clearly in the coarse mesh results. This can also be observed in the evolution of the dissipation of the kinetic energy, in Figure 6.2. The maximum dissipation time,

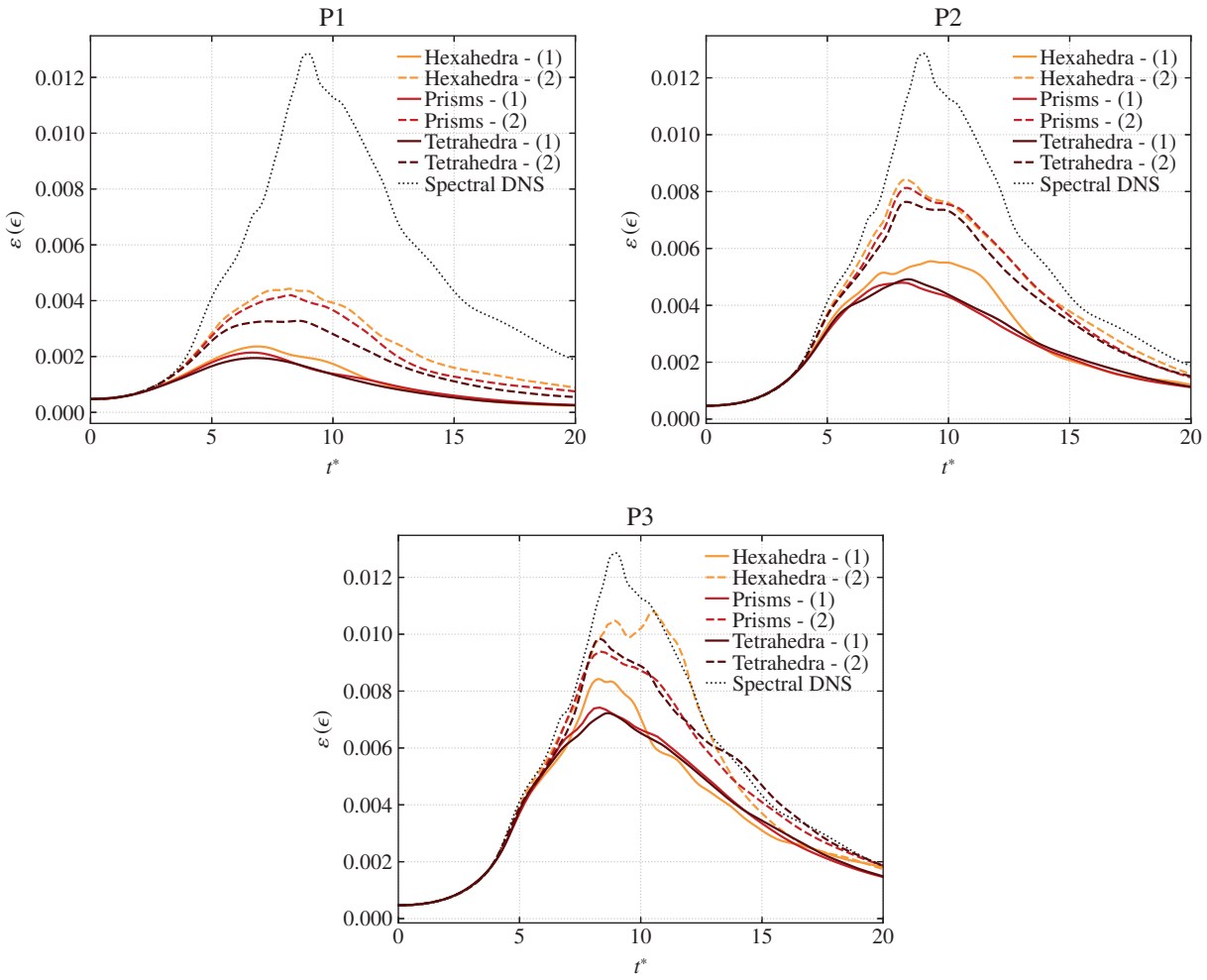


Figure 6.3. Temporal evolution of the dissipation rate based on the enstrophy integrated over the domain for $P1$-$P3$ schemes for the Taylor-Green vortex. Results are shown for both the coarse case (1) with 64^3 DOF and the finer case (2) with 128^3 DOF.

which occurs at around $t^*_{max} \approx 9$ for the DNS reference data, is poorly predicted by the lowest order schemes. Note that for all cases, increasing the polynomial degree substantially improves the results, and the maximum dissipation time from the numerical results approaches t^*_{max} for all element types.

Another perspective for the dissipation of the kinetic energy can be seen in terms of the enstrophy. For incompressible and weakly compressible flows, it is expected that the rate of decrease of energy is proportional to the enstrophy, calculated as

$$\epsilon = \frac{1}{\rho_0 \Omega} \int_\Omega \rho \frac{\boldsymbol{\zeta} \cdot \boldsymbol{\zeta}}{2} d\Omega. \tag{6.5}$$

where $\boldsymbol{\zeta}$ is the vorticity, ρ is the density and Ω is the volume of the computational domain. Figure 6.3

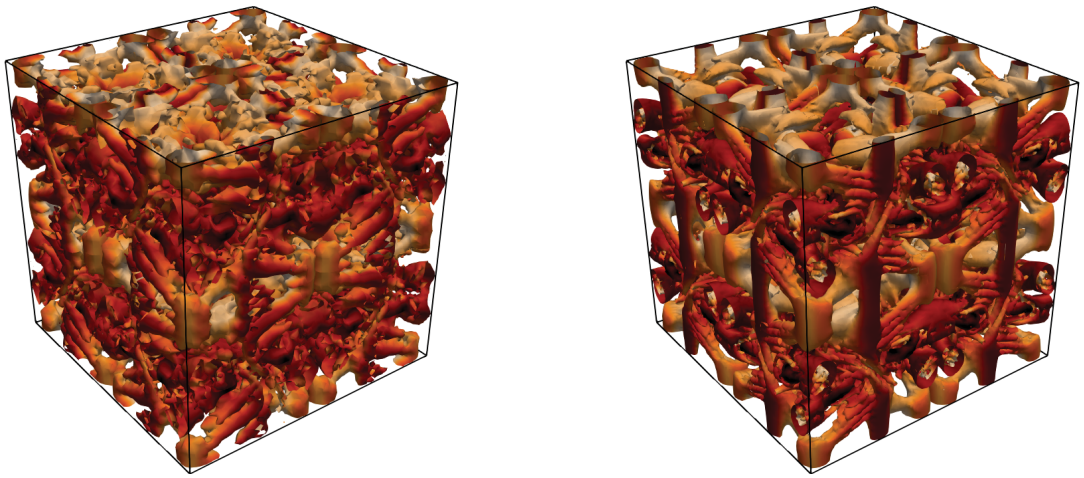
		Coarse		Fine	
		N_{c_i}	DOF_e/DOF_{hex}	N_{c_i}	DOF_e/DOF_{hex}
Hexahedra	P1	32	-	64	-
	P2	22	-	43	-
	P3	16	-	32	-
Prisms	P1	28	1.005	56	1.005
	P2	20	1.002	39	0.995
	P3	15	1.030	30	1.030
Tetrahedra	P1	23	1.114	45	1.043
	P2	17	1.025	33	1.004
	P3	13	1.006	26	1.006

Table 6.1. Number of cuboid elements per direction and ratio of total DOF compared to hexahedral computational domain for the Taylor-Green vortex.

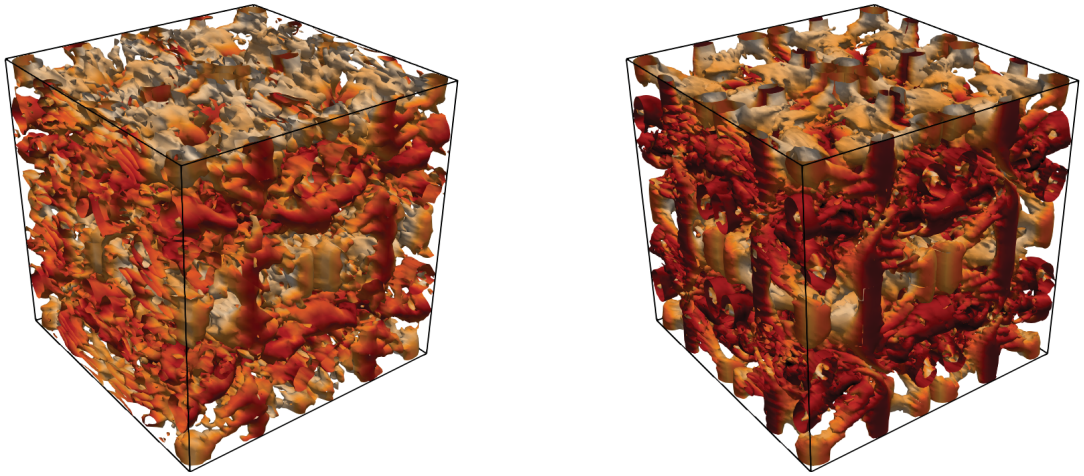
displays the evolution of the enstrophy in the domain. It is known that the maximum enstrophy is located at the smallest scales, so this serves as a measure of the performance of the implicit SGS model, which is simply defined by the grid and polynomial degree. The results for $P1$ display the unsuitability of this low-order spatial discretization for LES since most of the data is lost. Hence the amount of numerical error is large. We recall from our spectral analysis that hexahedral elements contain the smallest range of numerical error. The Taylor-Green vortex is characterized by its three-dimensional flow behaviour. Despite having computational grids with slightly more solution points for the prisms and tetrahedra in most simulations, as shown in Table 6.1, these two element types failed to outperform the hexahedral elements, consistent with the spectral analysis.

Finally, both dispersion and dissipation errors are visible in the turbulent structures in Figure 6.4, where isosurfaces of $Q \geqslant 1E-2$ are presented for the coarse and fine cases. The hexahedral elements display the turbulent structures with, qualitatively, more detail. The numerical error increases for the prisms and even more for the tetrahedral element. The latter displays quite large inter-element jumps with dispersed and damped modes. This noisy behaviour had been previously observed by Carton de Wiart [57]. In general, the details improve for all element types using the finer mesh. At this level, the amount of dissipation introduced is smaller. However, the pattern of accuracy and element types is observed again, with more detail in the turbulent structures.

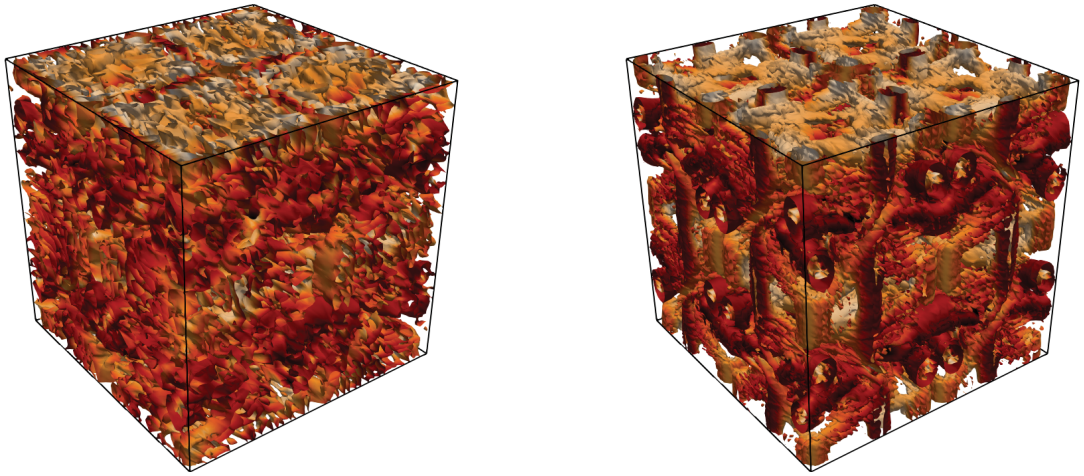
Hence, the results suggest that hexahedral elements are more accurate than tetrahedral and prismatic elements in three-dimensional turbulent flows using ILES.



(a) Hexahedra



(b) Prisms



(c) Tetrahedra

Figure 6.4. Isosurfaces of Q-criterion for the Taylor-Green vortex. Left side shows results for the coarser case (64^3 DOF) and right side for the fine case (128^3 DOF).

6.2 Turbulent Channel Flow

6.2.1 Case Description

Fully developed turbulent channel flow has been extensively studied to understand how turbulence behaves in wall-bounded flows [58]. DNS have been previously performed, and data is available [58] for validation of LES and ILES solvers. Turbulent channel flow consists of a flow driven by a pressure gradient dp/dx between two plates separated by a distance 2δ, where δ is the channel half width. The flow is governed by the centerline, bulk and friction Reynolds numbers

$$Re_c = \frac{\rho v_c \delta}{\mu}, \tag{6.6}$$

$$Re_b = \frac{\rho v_b 2\delta}{\mu}, \tag{6.7}$$

$$Re_\tau = \frac{\rho v_\tau \delta}{\mu}, \tag{6.8}$$

where ρ is the density of the fluid, μ is the dynamic viscosity, v_c and v_b are the centerline and the bulk velocity respectively, and $v_\tau = \sqrt{\tau_w/\rho}$ is the nondimensional wall friction velocity with τ_w the shear stress on the wall surface.

The flow was given initial conditions of $Re_b = 13600$ and a Mach number $M_a = 0.3$ to reach a friction velocity Reynolds number $Re_\tau = 395$. Sine and cosine functions and random perturbations Ψ were initially given to accelerate the turbulence transition

$$\frac{v_x}{v_b} = 2 + 0.147\left[\cos\left(\frac{y}{\delta}\right) + \sin\left(\frac{x}{\delta}\right) + \Psi\right], \tag{6.9}$$

$$\frac{v_y}{v_b} = 0.147\left[\cos\left(\frac{x}{\delta}\right) + \sin\left(\frac{y}{\delta}\right) + \Psi\right], \tag{6.10}$$

$$\frac{v_z}{v_b} = 0, \tag{6.11}$$

where the bulk velocity v_b is given by

$$v_b = \int_0^\delta \bar{v} d\left(\frac{y}{\delta}\right) \approx 6.8. \tag{6.12}$$

For this study, a coarse mesh with three-eighths the degrees of freedom per direction on average of that of Kim and Moser [58] was used in conjunction with $P1$, $P2$ and $P3$ schemes. A Rusanov Riemann solver was used for the common inviscid flux and the BR2 scheme for the common viscous fluxes. The total degrees of freedom for the ILES simulation is approximately one-twentieth of that required for DNS. The domain has dimensions $L_x \times L_y \times L_z = 2\pi\delta \times 2\delta \times \pi\delta$ with periodic boundary conditions in the streamwise and spanwise directions. The grid points in the y-direction were obtained with a hyperbolic function [59]

$$y_j = \frac{1}{2a} \tanh\left[\left(-1 + \frac{2j}{N}\right) \tanh^{-1} a\right] + 0.5, \tag{6.13}$$

where N is the number of grid points in the y-direction and $0 < a < 1$ is the stretching factor. For this study we used $a = 0.96$. We implemented the Gauss points for the hexahedra, the tensor product of Williams-Shunn [60] and Gauss points for prisms, and Shunn-Ham [61] points for the tetrahedra. Triangular and quadrilateral faces used the Williams-Shunn points and Gauss points, respectively. Table 6.2 shows the resulting number of degrees of freedom per direction for hexahedral meshes and the resulting nondimensional grid spacing. Keeping the number of elements fixed in the $y-$direction, the number of elements in the streamwise and spanwise direction were obtained keeping a relation of $4N_x/3N_z$ to maintain the aspect ratio of the elements. Table 6.3 shows the resulting number of elements for hexahedral, prismatic and tetrahedral elements for each polynomial degree.

Table 6.2. Number of degrees of freedom and nondimensional mesh spacing for hexahedral elements.

	$DOF(x \times y \times z)$	Δy^+	Δx^+	Δz^+
$P1$	$96 \times 72 \times 72$	0.819	25.852	17.235
$P2$	$96 \times 72 \times 72$	0.691	25.852	17.235
$P3$	$96 \times 72 \times 72$	0.599	25.852	17.235

Using the fourth-order Runge Kutta time-stepping scheme, each simulation was run using 120 cores until the friction velocity Reynolds number converged to 395 from $t^* \approx 816$ to $t^* \approx 950$ depending on the polynomial degree and element type, where t^* is normalized using the bulk velocity v_b and the channel half-width δ. Statistics were then collected over a period of $t^* = 136$, allowing about 20 passes of the flow through the channel. These statistics were also averaged in space to showcase the flow features in the wall-normal direction, and are displayed in Figures 6.5 to 6.7. Interestingly, the original plots obtained with the tetrahedral meshes presented scattering. To solve

Table 6.3. Number of cuboids N_c used for ILES of turbulent channel flow in each direction.

Element type		$N_x \times N_y \times N_z$
Hex	P1	$48 \times 36 \times 36$
	P2	$32 \times 24 \times 24$
	P3	$24 \times 18 \times 18$
Prism	P1	$40 \times 36 \times 30$
	P2	$28 \times 24 \times 21$
	P3	$22 \times 18 \times 17$
Tet	P1	$28 \times 36 \times 21$
	P2	$22 \times 24 \times 17$
	P3	$18 \times 18 \times 14$

this issue, the averaging in the normal direction was done with a higher range of values. This issue may be attributed to the irregular distribution of points in the spanwise-streamwise plane per y-DOF location for this element type.

6.2.2 Results

We first consider the plots of the streamwise velocity v_x^+ against y^+ in Figure 6.5. We observe good agreement with the results of Kim and Moser near the wall for all element types, but largely overpredicted values by the lower polynomial degree plots. This is a consequence of the increased dissipation of the $P1$ scheme as presented by Vermeire [14]. The hexahedral elements showed good agreement for the fourth-order spatial discretization, where the numerical results seem to capture the DNS data both close and far from the walls. Prisms and tetrahedra, on the other hand, still overpredict the streamwise velocity after $y^+ \approx 10$.

The root-mean-square velocity fluctuations $v_{x,rms}, v_{y,rms}$, and $v_{z,rms}$ intensities can be observed in Figure 6.6. Near the walls, all of the element types showed good agreement with the DNS results, after which overprediction of the normal components is evident. We observe that the overprediction issue is alleviated with increases of polynomial degree. We recall that hexahedral elements displayed the highest amount of dissipation for propagation normal to its faces. Despite the turbulent channel's mainly one-directional velocity field, this element type best represented the normal r.m.s fluctuations. We note that the increased DOF in the y-direction resulted in large reductions of the DOF in the spanwise and streamwise directions for the other two element types.

The $\overline{v_x' v_y'}$ component of the Reynolds stresses is plotted in Figure 6.7. We observe significant

underprediction in most of the domain for all element types for the lowest order schemes. Convergence to the DNS results is clearly seen as the polynomial order is increased. In all cases, the tetrahedral elements displayed the greatest amount of dissipation for the $\overline{v_x' v_y'}$ plots. Results for prisms and hexahedra were very close to each other, with significantly higher accuracy than the tetrahedra.

Figure 6.8 shows the isosurfaces of Q-criterion for hexahedral, prismatic and tetrahedral elements for $P3$. We see a slight difference in the results. The structures are more robust for the hexahedral mesh than for the other two. We pay particular attention to the shape of these structures for the tetrahedral mesh, which seems finer and more damped. This suggests a higher amount of numerical error, both dispersive and dissipative, for the tetrahedra and prisms compared to the hexahedra.

In general, the best performance can be attributed to the hexahedral elements, which agrees well with the findings from spectral analysis.

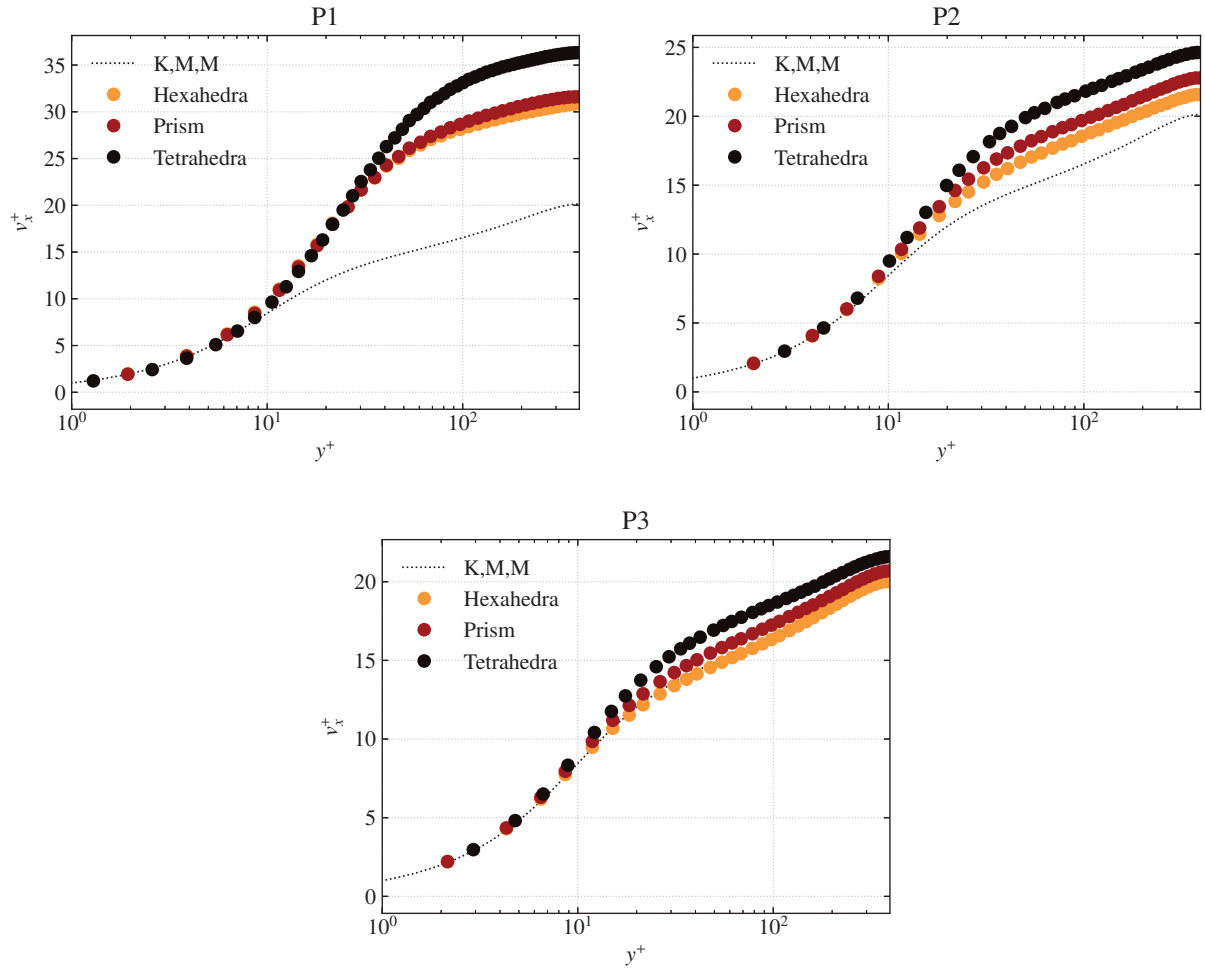


Figure 6.5. Mean velocity profile v_x^+ vs y^+ for the turbulent channel cases.

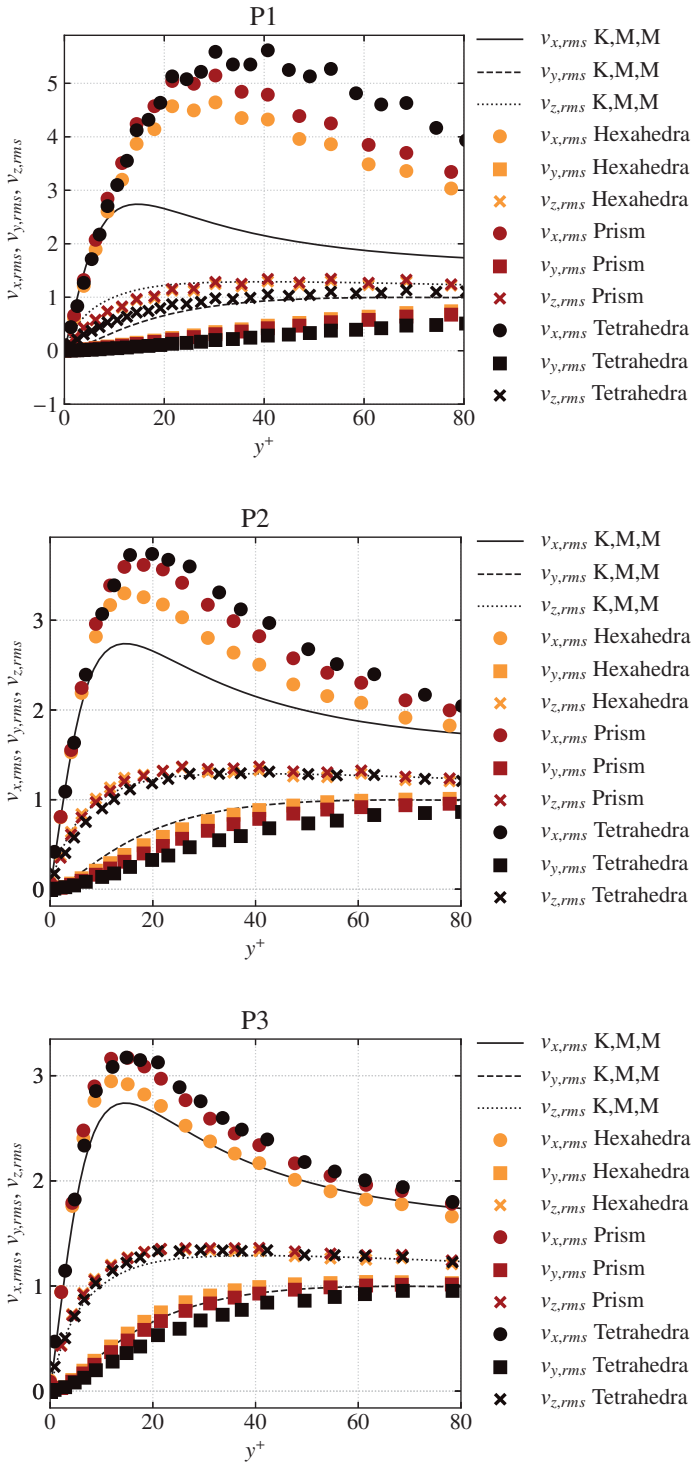


Figure 6.6. Root-mean square velocity fluctuations in wall coordinates for the turbulent channel cases.

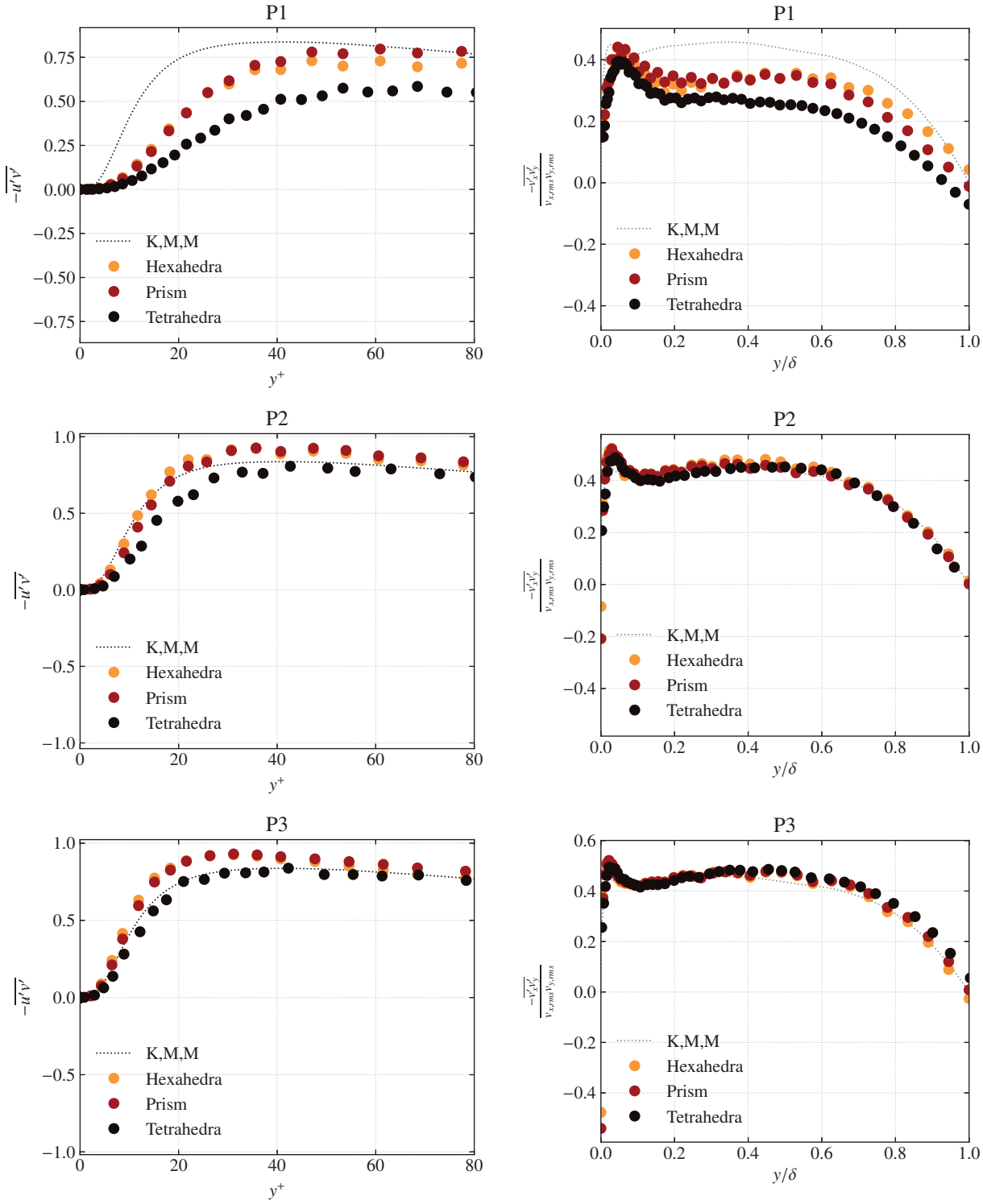


Figure 6.7. Reynolds stresses $\overline{v'_x v'_y}$ for the turbulent channel cases.

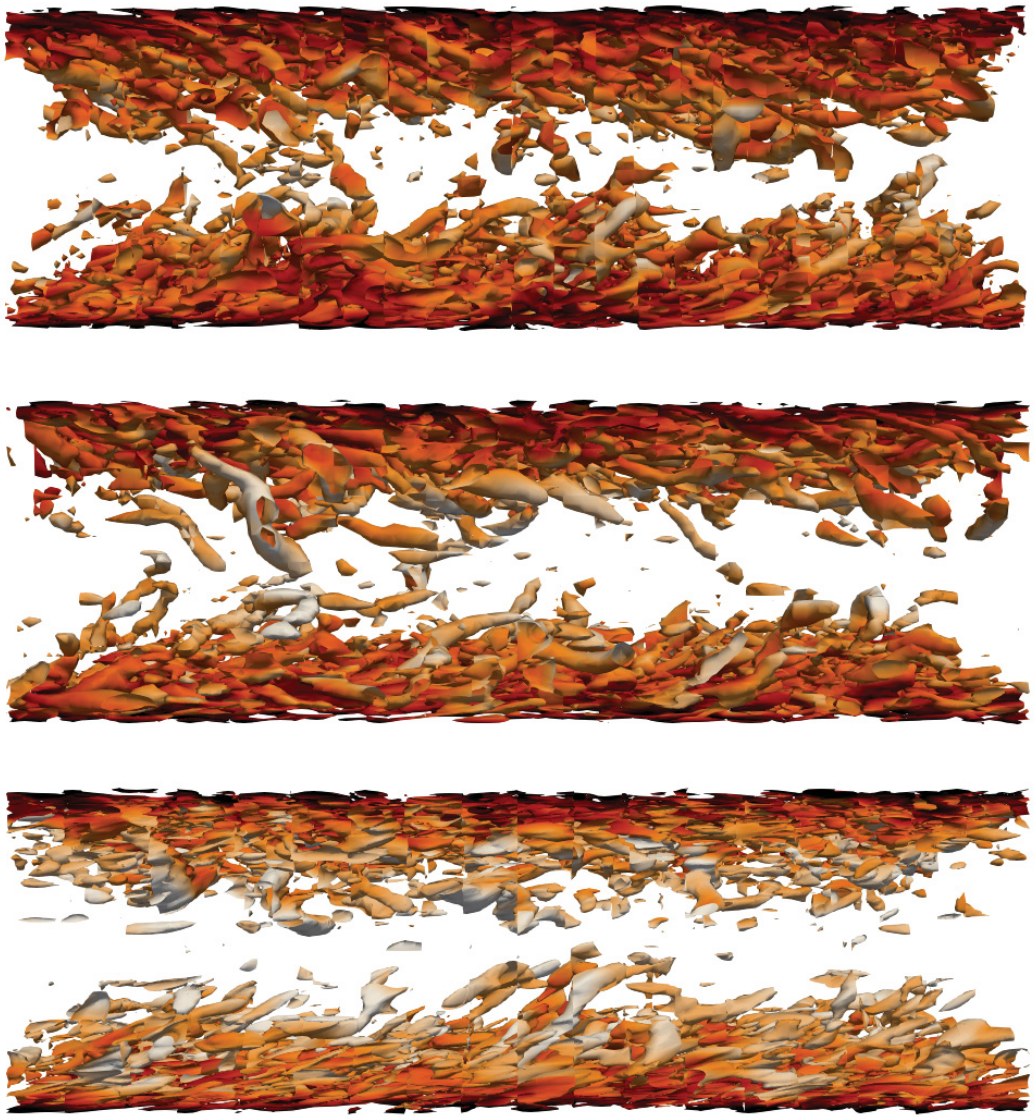


Figure 6.8. Isosurfaces of Q-criterion colored by the streamwise velocity for the turbulent channel using $P3$ for (a) hexahedra, (b), prisms, (c) tetrahedra.

6.3 Turbulent Flow over a Cylinder

6.3.1 Case Description

Turbulent flow over a cylinder has been the focus of study of many aerodynamicists due to its wide range of applications. The flow is dependent on the Reynolds number defined as

$$Re = \frac{\rho v_{x,\infty} d}{\mu}, \tag{6.14}$$

where ρ is the density of the fluid, $v_{x,\infty}$ is the free-stream velocity, d is the diameter and μ is the dynamic viscosity. We perform Large Eddy Simulation at $Re = 1000$, which according to Williamson [62] corresponds to the lower limit of the shear-layer transition regime. Computational domains with comparable numbers of points and generated using single element types were constructed for hexahedral, prismatic and tetrahedral elements. The orientation of the elements was chosen so that all the faces of the elements, in particular for the tetrahedra matched at the interfaces. This can be seen in Figure 6.9. In addition, the stretching ratio was maintained for all element types to ensure good resolution within the boundary layer. About 5 million degrees of freedom were used in the domain.

6.3.2 Results

The cases were run over $150t^*$, where $t^* = d/v_{x,\infty}$, as shown in Figure 6.10. As a first observation, we note a stronger circulation in the hexahedral and prismatic plots, where the values of the lift coefficient reach greater magnitudes compared to the tetrahedral elements. This is a first indication that the amount of numerical error may be higher for the tetrahedra.

Contours of the magnitude of the velocity can be seen in Figure 6.11. It is clear that the results for the hexahedral and prismatic elements are quite similar. In contrast, tetrahedral elements display inter-element jumps in the turbulent wake close to the low-pressure region behind the cylinder. We note that the mesh in this region was fine as shown in Figure 6.9, and yet the jumps across elements were visible for this element type.

We averaged the statistics during $25t^*$ to visualize the difference in the results among elements. In Figure 6.12 we observe the averaged streamwise velocity component at $x/d = 1$ and $x/d = 2$,

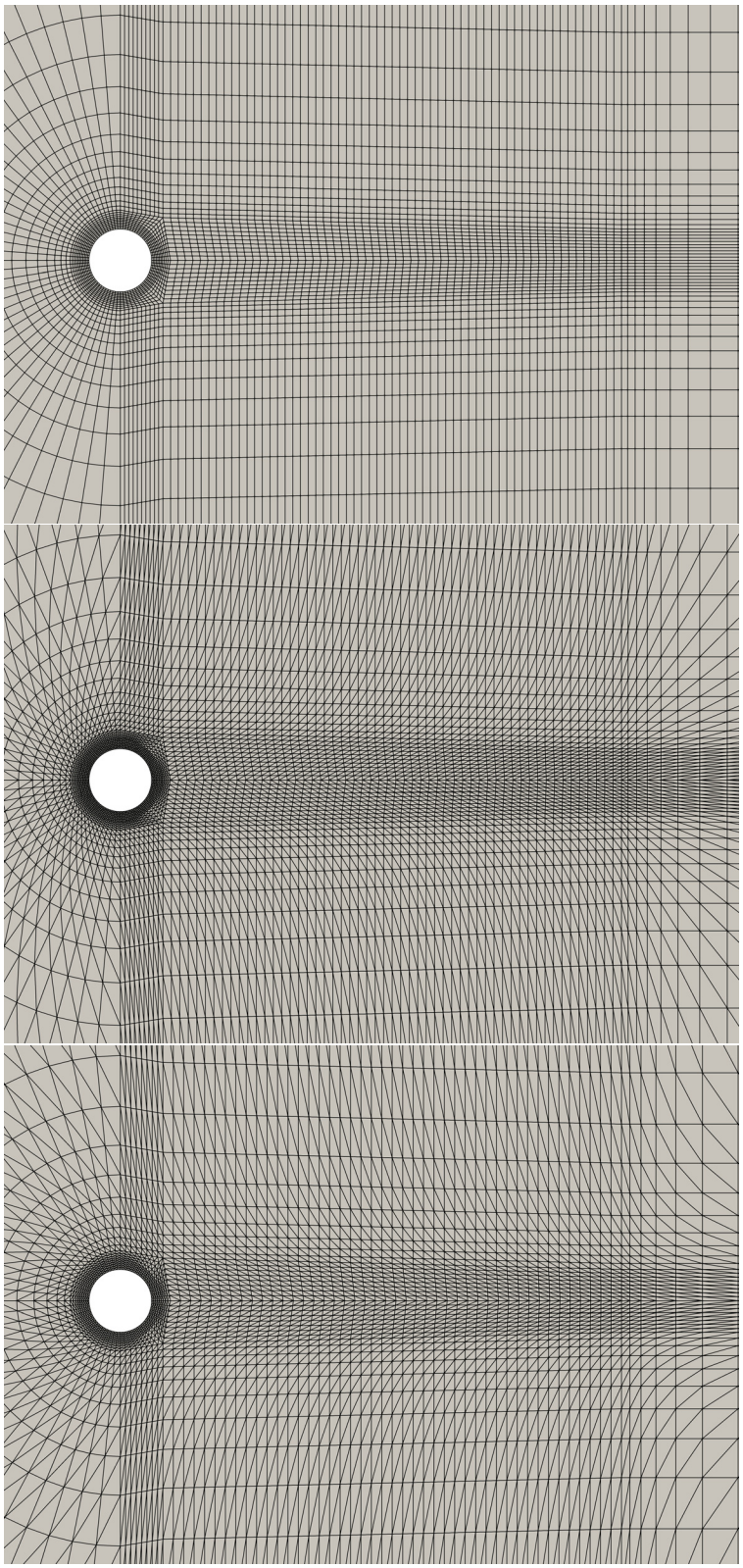


Figure 6.9. Hexahedral, prismatic and tetrahedral grids for the cylinder case.

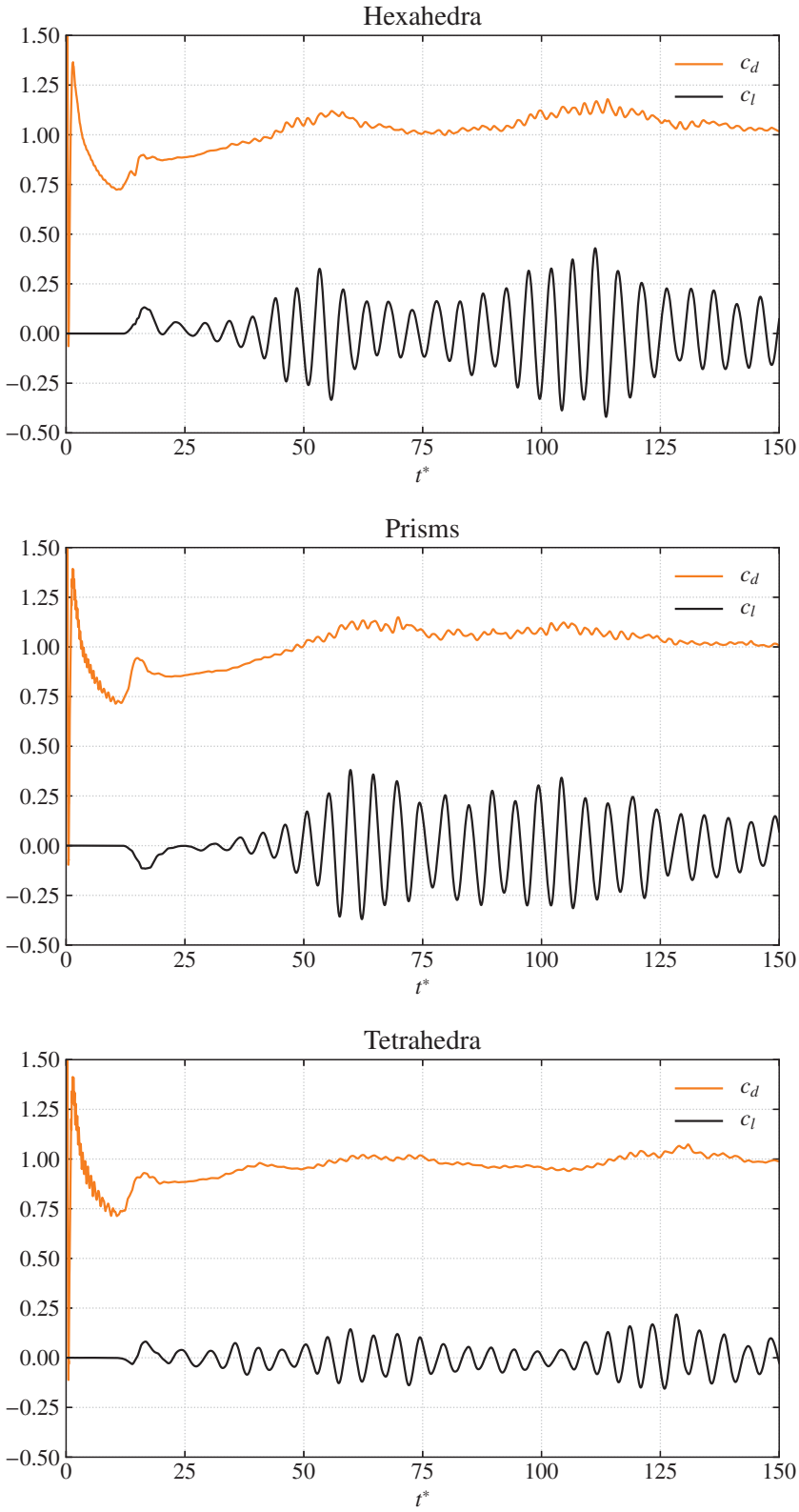


Figure 6.10. Lift and drag coefficients c_l and c_d evolution for the cylinder case.

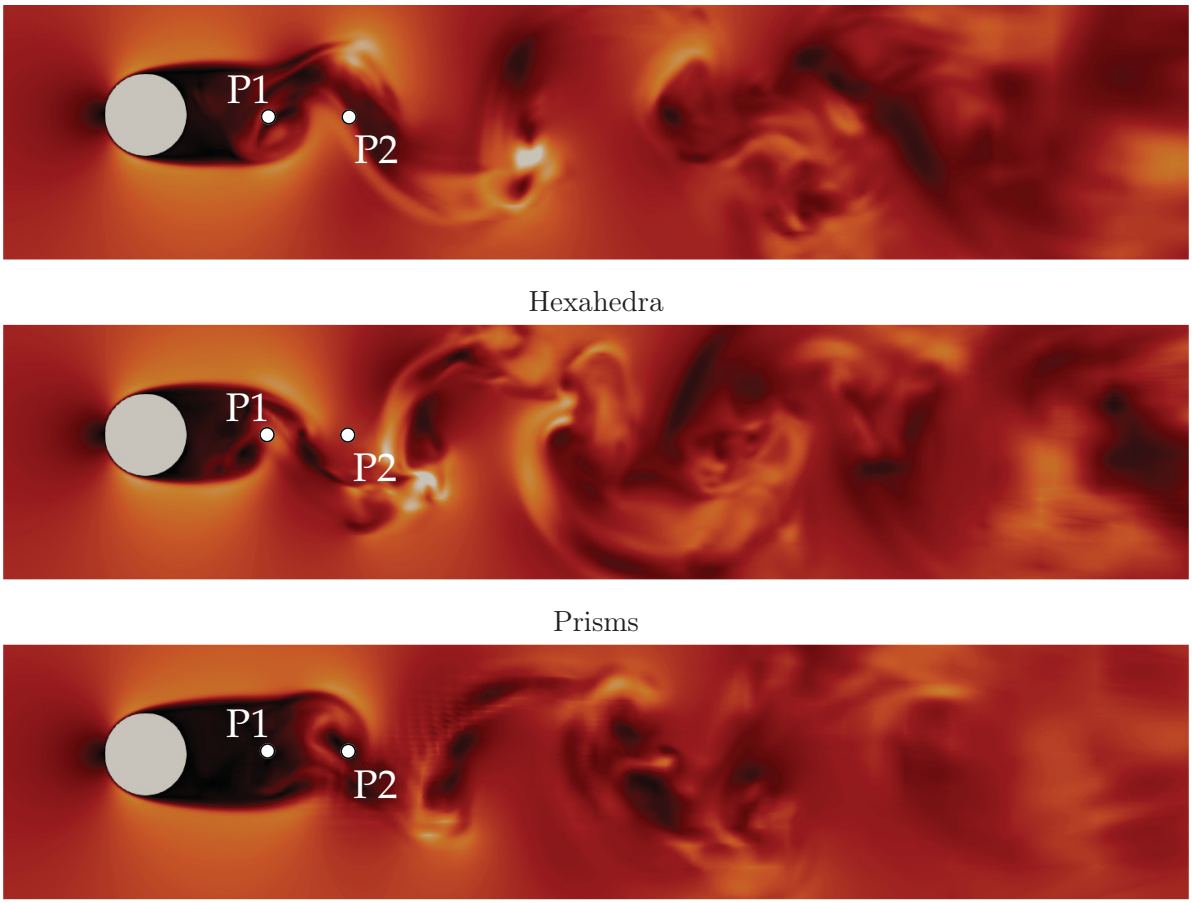


Figure 6.11. Contours of velocity magnitude for the cylinder case.

which are visually marked in Figure 6.11, and correspond to the x-coordinates of points $P1$ and $P2$, respectively. From the results of $x/d = 1$ in Figure 6.12, it appears that the time-averaged separation region behind the cylinder is larger for the tetrahedra. The values of the streamwise velocity close to the center of the cylinder are zero as a consequence of this. Next, we move further downstream by a factor of a diameter to the x-coordinate of point $P2$. Here, the averaged profiles show similar shapes, but again the results of tetrahedral grid stand out, since this x-location could still be within the recirculation bubble for these element types.

To visualize the averaged separation bubble, Figure 6.13 shows the averaged contours of the velocity magnitude. We observe that the tetrahedral results indeed have a slightly larger separation region compared to the other two element types, followed by the hexahedra and finally the prisms. This behaviour has been observed at different Reynolds numbers [63] and could be associated with a

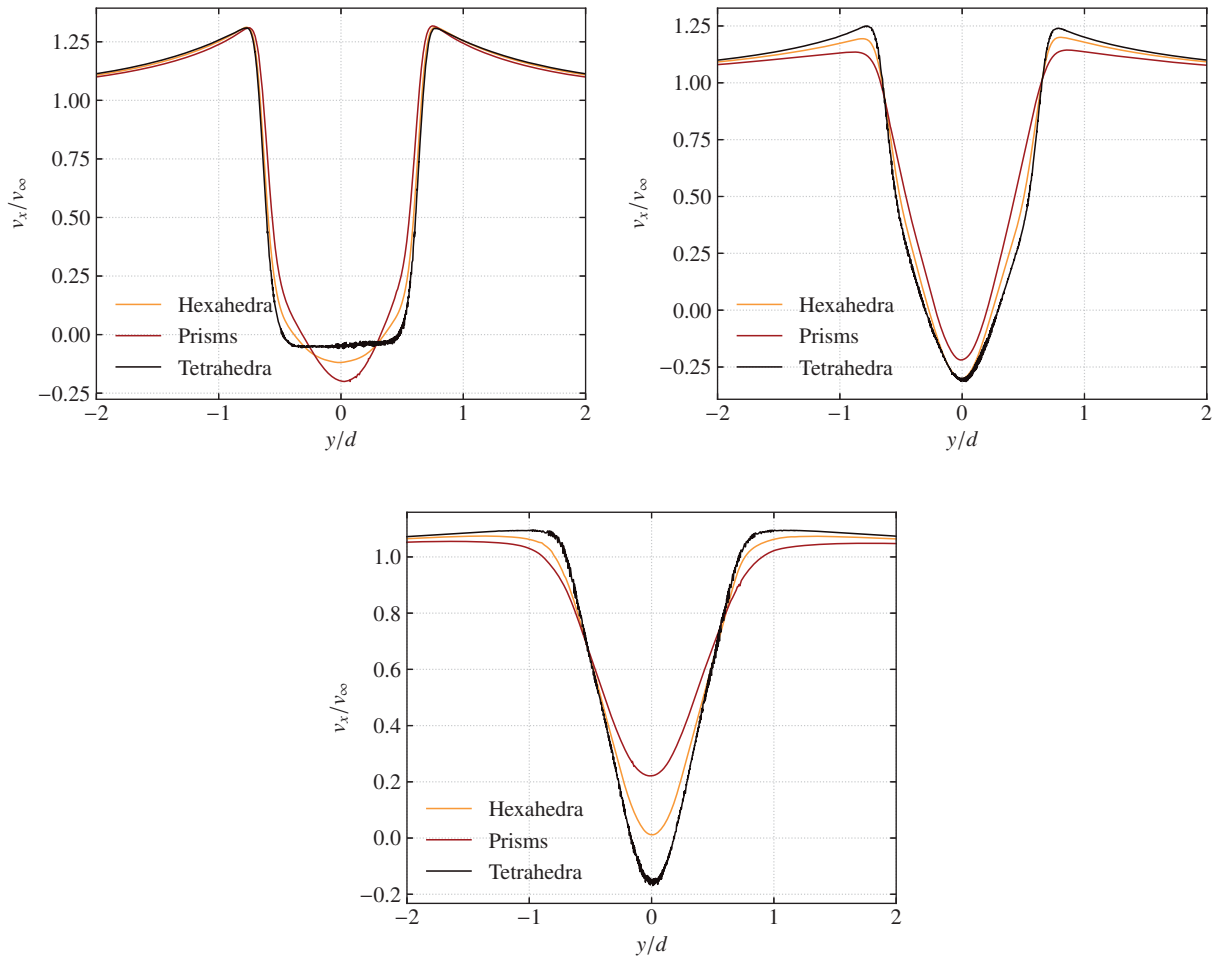
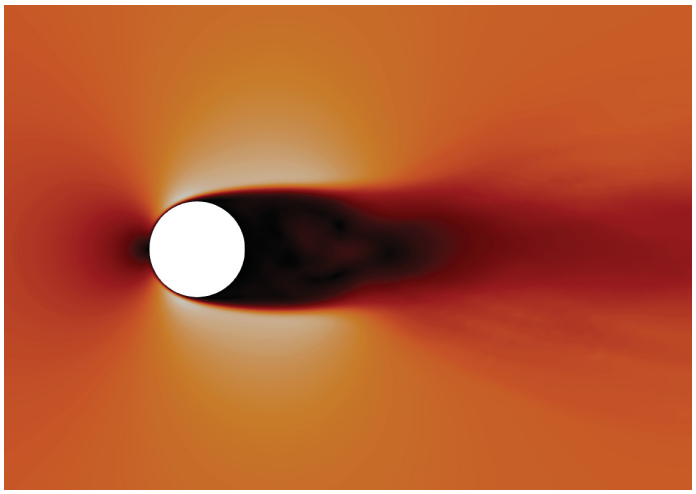


Figure 6.12. Mean velocity profiles for all element types at locations $x/d = [1.0, 1.5, 2.0]$ for the cylinder case.

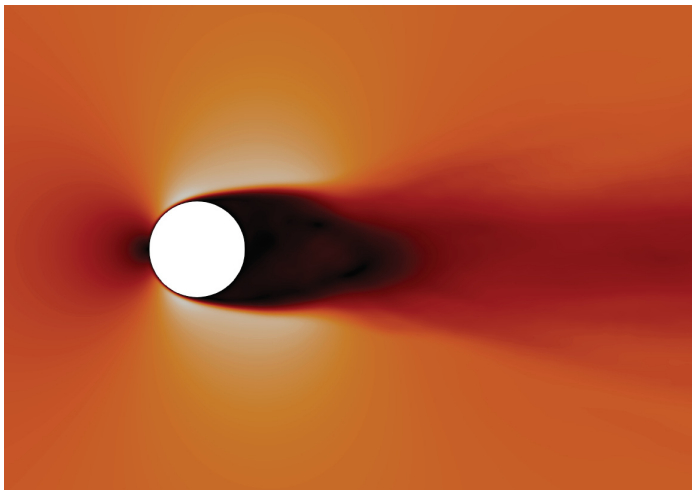
low-frequency physical mode, which has been previously documented by Lehmkuhl et al. [64].

The coefficient of drag averaged during $100t^*$ is shown in Table 6.4 and is compared with the DNS results from Zhao et al. [65]. In general, the results are quite close to the DNS results for all element types, with the hexahedra having the most accurate value, and similar to the prismatic results.

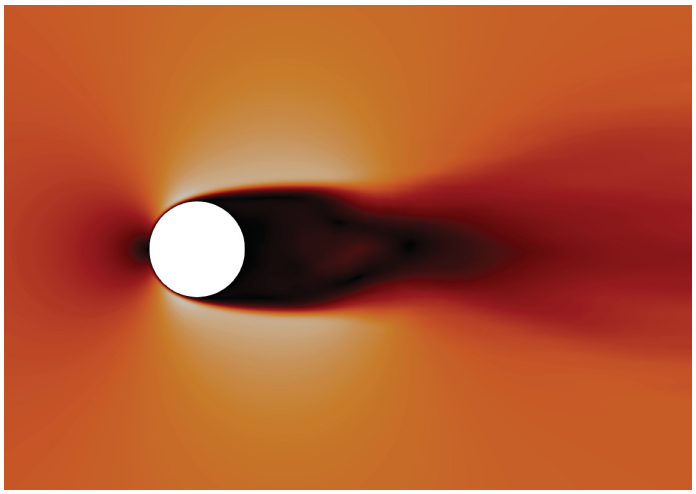
The noisy behaviour previously observed in the velocity magnitude contours can also be observed in the isosurfaces of Q-criterion displayed in Figure 6.14. We can see that hexahedral elements have the most robust turbulent structures compared to the other two element types. The results for both the turbulent wake close and far from the cylinder have generally a smooth behaviour. In the case of the prisms, the results were very good in the fine region, similar to the hexahedra's vortical



Hexahedra



Prisms



Tetrahedra

Figure 6.13. Time averaged contours of velocity magnitude the cylinder case.

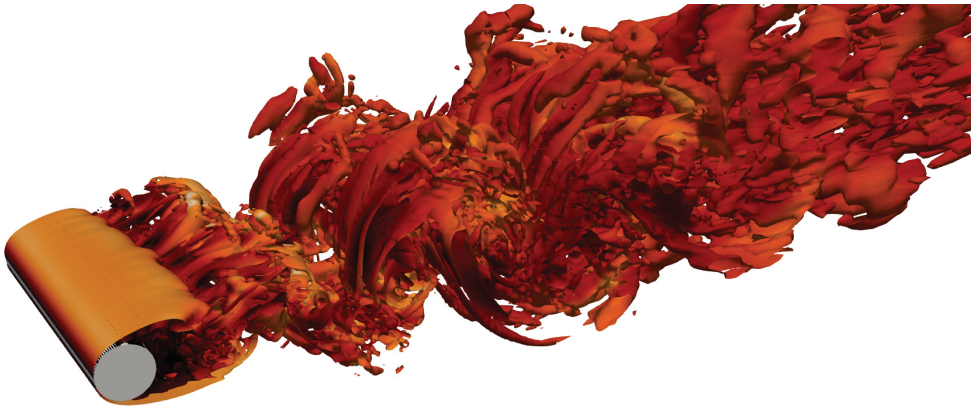
Table 6.4. Mean drag coefficient for hexahedral, prismatic and tetrahedral meshes for the cylinder case.

	$\overline{c_d}$
Zhao et al. (coarse)	1.092
Hexahedra	1.064
Prisms	1.062
Tetrahedra	0.991

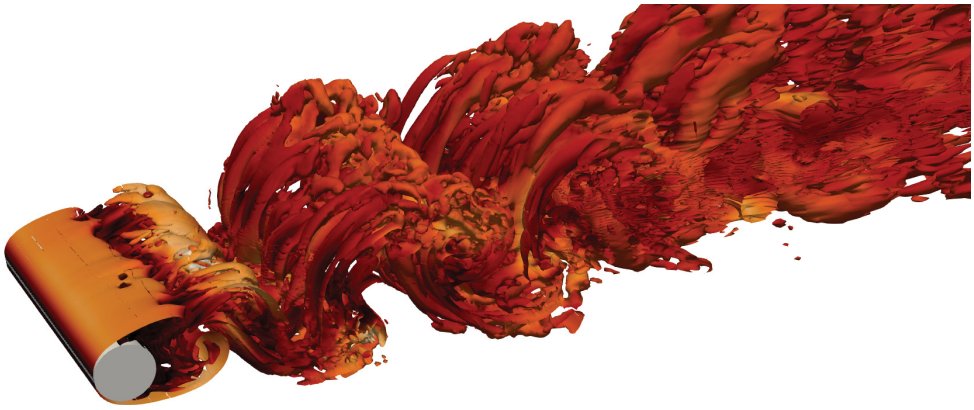
structures. However, far from the cylinder, some inter-element jumps become prominent. Finally, the tetrahedral grid results show noisy behaviour consisting of large solution jumps in the turbulent wake, even in the region close to the cylinder, where the mesh is relatively fine.

To visualize the energy and dissipation at different locations downstream of the cylinder, Figure 6.15 shows plots of the power spectral density (PSD) of the streamwise velocity fluctuations for all element types at $y/d = 0$ and $x/d = 1.79, 2.19, 3.38$ against the Strouhal number $S_t = \omega v_x/d$, where ω is the frequency. We note that the energy at low frequencies is comparable amongst the three element types. However, as we move to high-frequency regions the inherent dissipation of each element type starts to take effect. In particular, we observe the added dissipation of the prisms and most prominently from the tetrahedra, compared to the hexahedral semidiscretization. This is consistent with the contours in this section. Generally, we observe good behaviour from all element types in regions where the turbulent structures are large. As soon as we move into zones where the turbulent structures are smaller, the amount of numerical error added by the tetrahedral elements is the largest.

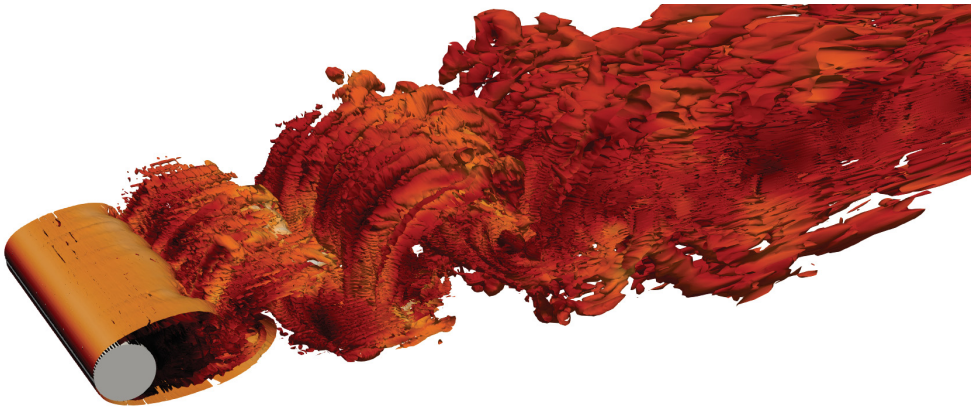
In closing, the results obtained with hexahedral and prismatic elements are generally good in the fine regions. As we move in the downstream direction, the results start to show more numerical error for the prisms and then the hexahedra. In the case of the tetrahedral elements, the numerical error was significantly higher for the current mesh resolution.



Hexahedra



Prisms



Tetrahedra

Figure 6.14. Isosurfaces of Q-criterion for the cylinder case.

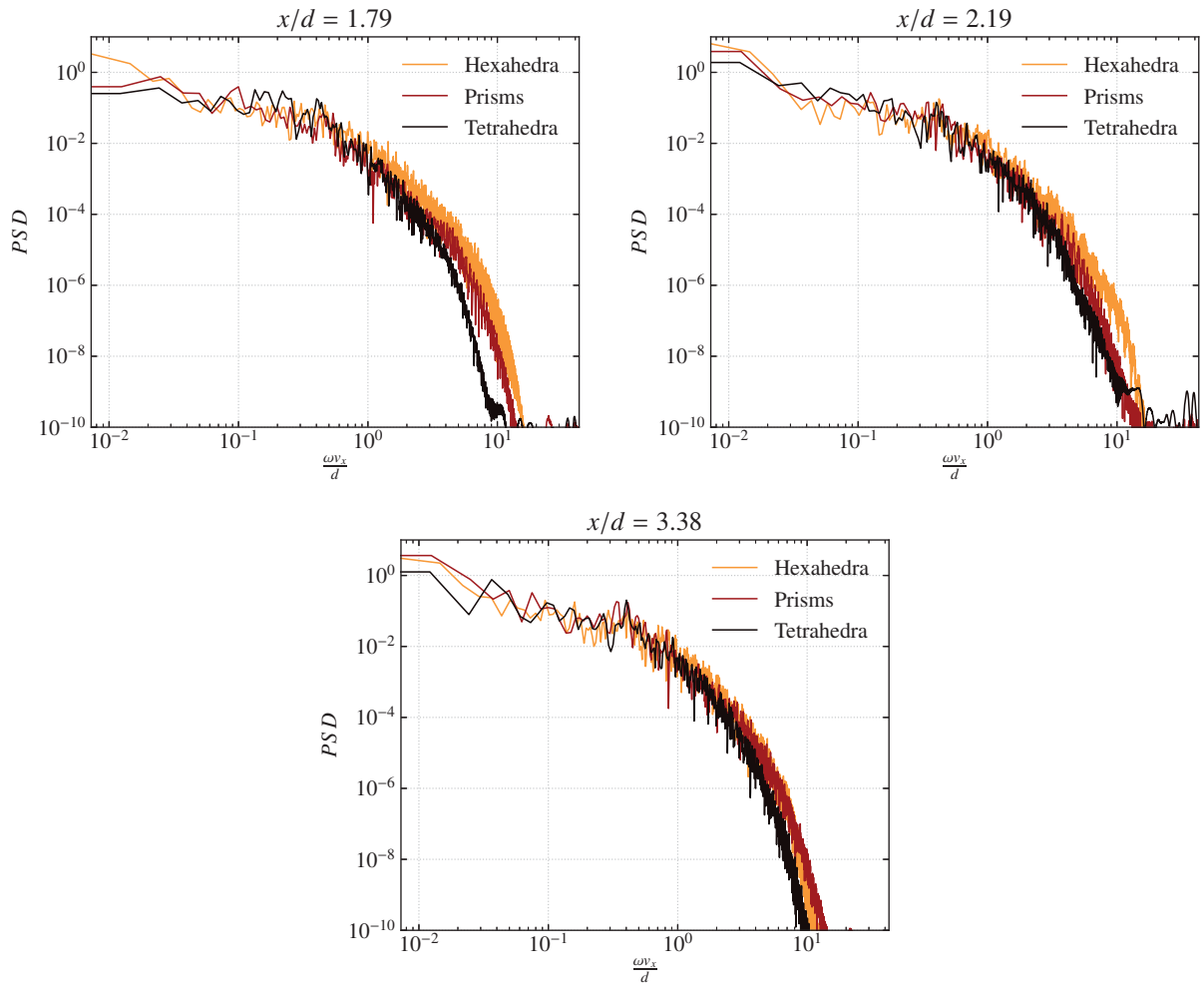


Figure 6.15. Power Density Spectra for the cylinder case at different x/d locations against the Strouhal number.

Chapter 7

Conclusions and Future Work

In this work, we have described several procedures that serve to analyze the behaviour of element types in semidiscrete systems. In closing this work, we outline some of the findings and summarize the results obtained throughout.

We have studied the characteristic error for spatial discretizations composed of hexahedral, prismatic and tetrahedral elements. We made use of the implicit large eddy simulation turbulence modelling approach. It is known that FR provides effective modelling of the unresolved dynamics, and that no subgrid-scale turbulence models are required. Instead, FR relies on the numerical dissipation of the spatial scheme to model the smallest scales in turbulent flows at high Reynolds numbers.

First, we verified the inviscid and viscous flux implementations to support our spectral analysis and numerical experiments. The results converged to the expected order of accuracy, which has been generally observed to be $P + 1$ in logarithmic scale for FR schemes. Next, we described the three-dimensional procedure of the spectral analysis. From this analysis, the numerical dispersion and dissipation errors were evaluated based on the exact dispersion relation. A set of $N_{DOF}N_{sub}$ eigenvalues were analyzed for all wavenumbers, and it was explained that the dominant mode contained the least dissipation at low wavenumbers. Despite more than one possible physical mode at high wavenumbers, a single eigenvalue dominates the wave propagation properties. This mode was observed to be quasi-periodic in three dimensions, compared to the periodicity observed in one and two dimensions in previous research.

Changing the orientation of the wave vector away from the orthogonal directions substantially changed the dispersion curves, the numerical dispersion and dissipation displayed non-smooth behaviour and jumps for the prisms and more notably for the tetrahedra. Hence, changing the wave orientation may have a significant impact on the scheme's numerical error. Contour plots of the

dispersion and dissipation error for a given wavenumber display this dependence of the error on the direction. In particular, elements with triangular faces show the least error normal to the triangle's diagonal, whereas the hexahedral element performs best in diagonal directions. We followed the analysis with a comparison of all element types normalizing the number of degrees of freedom of the periodic generating patterns. This showed that hexahedral elements are the least dissipative and dispersive amongst three element types for ILES, followed by the prisms and finally the tetrahedra. Not only do the hexahedral elements contain the least dissipation but also the range of error as the angle changes stay within a smaller range than it does for the other two element types.

Conclusions from this spectral analysis were then validated with turbulent cases including the Taylor-Green vortex, turbulent channel flow at $Re_\tau = 395$ and turbulent flow over a cylinder to test how well different element types resolve turbulence in the context of ILES. We implemented computational domains with comparable numbers of degrees of freedom. Hence, contrasting element types was done on a per degree of freedom basis. In most scenarios, the hexahedral elements displayed the most accurate results. The amount of dissipation added to the scheme was sufficient to resolve the turbulent structures yet provide robust simulations until at least $P3$. Prismatic and tetrahedral elements, damped the turbulent structures faster, meaning that the observed amount of numerical error was considerably higher. Isosurfaces of the Q-criterion visually supported the conclusions. In all ILES simulations, the turbulent structures were observed to be more robust and detailed for the hexahedral elements, while inter-element jumps were visible for the other two element types, regardless of the refinement level used, in particular for the tetrahedral elements.

In the future, we expect to analyze the behaviour of these three-dimensional semidiscretizations when coupled with explicit conventional time-marching schemes, as well as with the new optimized P-ERK schemes by Vermeire [66]. In addition, we will look into entropy stability and its application to hexahedra, prismatic and tetrahedral elements.

References

[1] Bernardo Cockburn and Chi-Wang Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Mathematics of computation*, 52(186):411–435, 1989.

[2] Bernardo Cockburn, San-Yih Lin, and Chi-Wang Shu. TBV Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *Journal of Computational Physics*, 84(1):90–113, 1989.

[3] Bernardo Cockburn, Suchung Hou, and Chi-Wang Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case. *Mathematics of Computation*, 54(190):545–581, 1990.

[4] Zhi Jian Wang. Spectral (finite) volume method for conservation laws on unstructured grids. basic formulation: Basic formulation. *Journal of computational physics*, 178(1):210–251, 2002.

[5] Yen Liu, Marcel Vinokur, and ZJ Wang. Discontinuous spectral difference method for conservation laws on unstructured grids. In *Computational Fluid Dynamics 2004*, pages 449–454. Springer, 2006.

[6] H.T. Huynh, Z Wang, and P.E. Vincent. High-Order Methods for Computational Fluid Dynamics: A Brief Review of Compact Differential Formulation on Unstructured Grids. *Computers & Fluids*, 98, June 2013.

[7] John A. Ekaterinaris. High-order accurate, low numerical diffusion methods for aerodynamics. *Progress in Aerospace Sciences*, 41(3):192–300, April 2005.

[8] Jeffrey Slotnick, Abdollah Khodadoust, Juan Alonso, David Darmofal, William Gropp, Elizabeth Lurie, and Dimitri Mavriplis. CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences. page 58.

[9] Hung T Huynh. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In *18th AIAA Computational Fluid Dynamics Conference*, page 4079, 2007.

[10] Zhi Jian Wang and Haiyang Gao. A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids. *Journal of Computational Physics*, 228(21):8161–8186, 2009.

[11] T Haga, H Gao, and Zhi Jian Wang. A high-order unifying discontinuous formulation for the Navier-Stokes equations on 3D mixed grids. *Mathematical Modelling of Natural Phenomena*, 6(3):28–56, 2011.

[12] DM Williams and Antony Jameson. Energy stable flux reconstruction schemes for advection–diffusion problems on tetrahedra. *Journal of Scientific Computing*, 59(3):721–759, 2014.

[13] Brian C Vermeire, Siva Nadarajah, and Paul G Tucker. Implicit large eddy simulation using the high-order correction procedure via reconstruction scheme. *International Journal for Numerical Methods in Fluids*, 82(5):231–260, 2016.

[14] Brian Vermeire, Jean-Sebastien Cagnone, and Sivakumaran Nadarajah. ILES using the correction procedure via reconstruction scheme. In *51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 1001, 2013.

[15] Rodrigo C Moura, SJ Sherwin, and Joaquim Peiró. Linear dispersion–diffusion analysis and its application to under-resolved turbulence simulations using discontinuous Galerkin spectral/hp methods. *Journal of Computational Physics*, 298:695–710, 2015.

[16] Rodrigo C Moura, SJ Sherwin, and Joaquim Peiró. Eigensolution analysis of spectral/hp continuous Galerkin approximations to advection–diffusion problems: Insights into spectral vanishing viscosity. *Journal of Computational Physics*, 307:401–422, 2016.

[17] RC Moura, J Peiró, and SJ Sherwin. Implicit LES Approaches via Discontinuous Galerkin Methods at Very Large Reynolds. In *Direct and Large-Eddy Simulation XI*, pages 53–59. Springer, 2019.

[18] Peter E Vincent, Patrice Castonguay, and Antony Jameson. Insights from von Neumann analysis of high-order flux reconstruction schemes. *Journal of Computational Physics*, 230(22):8134–8154, 2011.

[19] Kartikey Asthana and Antony Jameson. High-order flux reconstruction schemes with minimal dispersion and dissipation. *Journal of Scientific Computing*, 62(3):913–944, 2015.

[20] B. C. Vermeire and P. E. Vincent. On the behaviour of fully-discrete flux reconstruction schemes. *Computer Methods in Applied Mechanics and Engineering*, 315:1053–1079, March 2017.

[21] Kris Van den Abeele. Development of high-order accurate schemes for unstructured grids. *Phd thesis in Vrije Universiteit Brussel*, 2009.

[22] Patrice Castonguay, Peter E Vincent, and Antony Jameson. A new class of high-order energy stable flux reconstruction schemes for triangular elements. *Journal of Scientific Computing*, 51(1):224–256, 2012.

[23] Fang Q Hu, MY Hussaini, and Patrick Rasetarinera. An analysis of the discontinuous Galerkin method for wave propagation problems. *Journal of Computational Physics*, 151(2):921–946, 1999.

[24] Will Trojak, Rob Watson, Ashley Scillitoe, and Paul G Tucker. Effect of Mesh Quality on Flux Reconstruction in Multi-Dimensions. *arXiv preprint arXiv:1809.05189*, 2018.

[25] Kris Van den Abeele, Ghader Ghorbaniasl, Matteo Parsani, and Chris Lacor. A stability analysis for the spectral volume method on tetrahedral grids. *Journal of Computational Physics*, 228:257–265, 2009.

[26] Peter Vincent, Freddie Witherden, B.C. Vermeire, Jin Seok Park, and Arvind Iyer. Towards Green Aviation with Python at Petascale. November 2016.

[27] S. B. Pope and Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, August 2000.

[28] Peter Davidson. *Turbulence: An Introduction for Scientists and Engineers*. Oxford University Press, 2015.

[29] Paul Tucker. *Unsteady Computational Fluid Dynamics in Aeronautics*. Fluid Mechanics and Its Applications. Springer Netherlands, 2014.

[30] Pierre Sagaut. Large Eddy Simulation for Incompressible Flows - An Introduction | P. Sagaut | Springer. https://www.springer.com/gp/book/9783540263449.

[31] Dimitris Drikakis, Christer Fureby, Fernando F. Grinstein, and David Youngs. Simulation of transition and turbulence decay in the Taylor–Green vortex. *Journal of Turbulence*, 8:N20, January 2007.

[32] Patrick J. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, 1998.

[33] Guide: Guide for the Verification and Validation of Computational Fluid Dynamics Simulations (AIAA G-077-1998(2002)) | AIAA Standards. https://arc.aiaa.org/doi/10.2514/4.472855.001.

[34] Ivo Babuska and J.Tinsley Oden. Verification and validation in computational engineering and science: Basic concepts. *Computer Methods in Applied Mechanics and Engineering*, 193(36-38):4057–4066, September 2004.

[35] William L Oberkampf and Timothy G Trucano. Verification and Validation in Computational Fluid Dynamics. page 122.

[36] Guide: Guide for the Verification and Validation of Computational Fluid Dynamics Simulations (AIAA G-077-1998(2002)). In *Guide: Guide for the Verification and Validation of Computational Fluid Dynamics Simulations (AIAA G-077-1998(2002))*, AIAA Standards. American Institute of Aeronautics and Astronautics, Inc., January 1998.

[37] Farshad Navah. Development, Verification and Validation of High-Order Methods for the Simulation of Turbulence. page 284.

[38] Verification and validation in computational science and engineering - CERN Document Server. http://cds.cern.ch/record/580994.

[39] Zhijian J Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, Hung T Huynh, et al. High-order

CFD methods: Current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[40] Manuel López Morales, Jonathan Bull, Jacob Crabill, Thomas Economon, David Manosalvas-Kjono, Joshua Romero, Abhishek Sheshadri, Jerry Watkins, David M. Williams, Francisco Palacios, and Antony Jameson. Verification and Validation of HiFiLES: A High-Order LES unstructured solver on multi-GPU platforms. In *32nd AIAA Applied Aerodynamics Conference*, June 2014.

[41] Brian C Vermeire, Freddie D Witherden, and Peter E Vincent. On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools. *Journal of Computational Physics*, 334:497–521, 2017.

[42] Charles Hirsch. *Numerical Computation of Internal and External Flows*. Elsevier, 2007.

[43] H. T. Huynh. A Reconstruction Approach to High-Order Schemnes Including Discontinuous Galerkin for Diffusion. In *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics.

[44] Bernardo Cockburn, George E. Karniadakis, and Chi-Wang Shu. The Development of Discontinuous Galerkin Methods. In Bernardo Cockburn, George E. Karniadakis, and Chi-Wang Shu, editors, *Discontinuous Galerkin Methods*, Lecture Notes in Computational Science and Engineering, pages 3–50. Springer Berlin Heidelberg, 2000.

[45] Francesco Bassi and Stefano Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of computational physics*, 131(2):267–279, 1997.

[46] F. Bassi and S. Rebay. A High Order Discontinuous Galerkin Method for Compressible Turbulent Flows. In Bernardo Cockburn, George E. Karniadakis, and Chi-Wang Shu, editors, *Discontinuous Galerkin Methods*, Lecture Notes in Computational Science and Engineering, pages 77–88. Springer Berlin Heidelberg, 2000.

[47] Wilhelm Kutta. *Beitrag zur näherungsweisen Integration totaler Differentialgleichungen*. Teubner, 1901.

[48] Roger Alexander. Diagonally Implicit Runge–Kutta Methods for Stiff O.D.E.'s. *Siam Journal on Numerical Analysis - SIAM J NUMER ANAL*, 14:1006–1021, December 1977.

[49] Brian Vermeire. Adaptive implicit-explicit time integration and high-order unstructured methods for implicit large Eddy simulation. 2014.

[50] John D. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill Education, New York, 5 edition edition, February 2010.

[51] Hung T Huynh. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In *18th AIAA Computational Fluid Dynamics Conference*, page 4079, 2007.

[52] Brian C Vermeire and Peter E Vincent. On the properties of energy stable flux reconstruction schemes for implicit large eddy simulation. *Journal of Computational Physics*, 327:368–388, 2016.

[53] Pablo Fernandez, Rodrigo C. Moura, Gianmarco Mengaldo, and Jaime Peraire. Non-modal analysis of spectral element methods: Towards accurate and robust large-eddy simulations. *Computer Methods in Applied Mechanics and Engineering*, 346:43–62, April 2019.

[54] Kris Van den Abeele. Development of high-order accurate schemes for unstructured grids. *Phd thesis in Vrije Universiteit Brussel*, 2009.

[55] Rodrigo C Moura, SJ Sherwin, and Joaquim Peiró. Eigensolution analysis of spectral/hp continuous Galerkin approximations to advection–diffusion problems: Insights into spectral vanishing viscosity. *Journal of Computational Physics*, 307:401–422, 2016.

[56] Zhijian J Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, Hung T Huynh, et al. High-order CFD methods: Current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[57] C. Carton de Wiart, K. Hillewaert, M. Duponcheel, and G. Winckelmans. Assessment of a discontinuous Galerkin method for the simulation of vortical flows at high Reynolds number. *International Journal for Numerical Methods in Fluids*, 74(7):469–493, 2014.

[58] John Kim, Parviz Moin, and Robert Moser. Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of fluid mechanics*, 177:133–166, 1987.

[59] Hiroyuki Abe, Hiroshi Kawamura, and Yuichi Matsuo. Direct numerical simulation of a fully developed turbulent channel flow with respect to the Reynolds number dependence. *Journal of fluids Engineering*, 123(2):382–393, 2001.

[60] DM Williams, Lee Shunn, and Antony Jameson. Symmetric quadrature rules for simplexes based on sphere close packed lattice arrangements. *Journal of Computational and Applied Mathematics*, 266:18–38, 2014.

[61] Lee Shunn and Frank Ham. Symmetric quadrature rules for tetrahedra based on a cubic close-packed lattice arrangement. *Journal of Computational and Applied Mathematics*, 236(17):4348–4364, 2012.

[62] C H K Williamson. Vortex Dynamics in the Cylinder Wake. *Annual Review of Fluid Mechanics*, 28(1):477–539, 1996.

[63] C H K Williamson. Vortex Dynamics in the Cylinder Wake. *Annual Review of Fluid Mechanics*, 28(1):477–539, 1996.

[64] Oriol Lehmkuhl, Ivette Rodriguez, R Borrell, carlos-david perez-segarra, and Assensi Oliva. Low-frequency variations in the wake of a circular cylinder at Re = 3900. *Journal of Physics: Conference Series*, 318:042038, December 2011.

[65] Ming Zhao, Liang Cheng, and Tongming Zhou. Direct numerical simulation of three-dimensional flow past a yawed circular cylinder of infinite length. *Journal of Fluids and Structures*, 25(5):831–847, July 2009.

[66] Brian C. Vermeire. Paired explicit Runge-Kutta schemes for stiff systems of equations. *Journal of Computational Physics*, 393:465–483, September 2019.

[67] 25.3. unittest — Unit testing framework — Python 2.7.16 documentation. https://docs.python.org/2/library/unittest.html.

[68] Yet Another Markup Language (YAML) 1.0. https://yaml.org/spec/history/2001-12-10.html.

Appendix A

Concentration of Dominant and Secondary Eigencurves

In this section, we show the resulting eigencurves for hexahedra, prisms and tetrahedral elements considering both the spurious and physical modes. Given the high density of these plots, only the results for the second-order spatial discretization ($P1$ schemes) are shown (See Chapter 5). As explained in this work, the dominant mode is considered to be that which has the least amount of numerical dissipation in the range of resolved wavenumbers and is represented using continuous lines for multiple wave orientations. We only considered waves aligned with the advection direction. Figure A.1 shows the concentration of modes for hexahedral elements. The shapes of these modes are smooth and a pattern can be observed. If we compare these results to those of the prismatic elements in Figure A.2, it is clear that the prisms yield less smooth dispersion curves. The consequence of the nonsmooth behaviour is still not well understood, but it is possibly a cause of increased numerical error as pointed out by Fernandez et al. [53]. Even less smooth behaviour is shown if we observe the results for the tetrahedral semidiscretizations in Figure A.3. This element type was concluded in the present work to be the least accurate of the three element types. The curves that result from tetrahedral spatial schemes do not follow specific behaviours such as the one observed in the tetrahedral results. What becomes even more interesting is that at given wave orientations, the numerical error increases vastly.

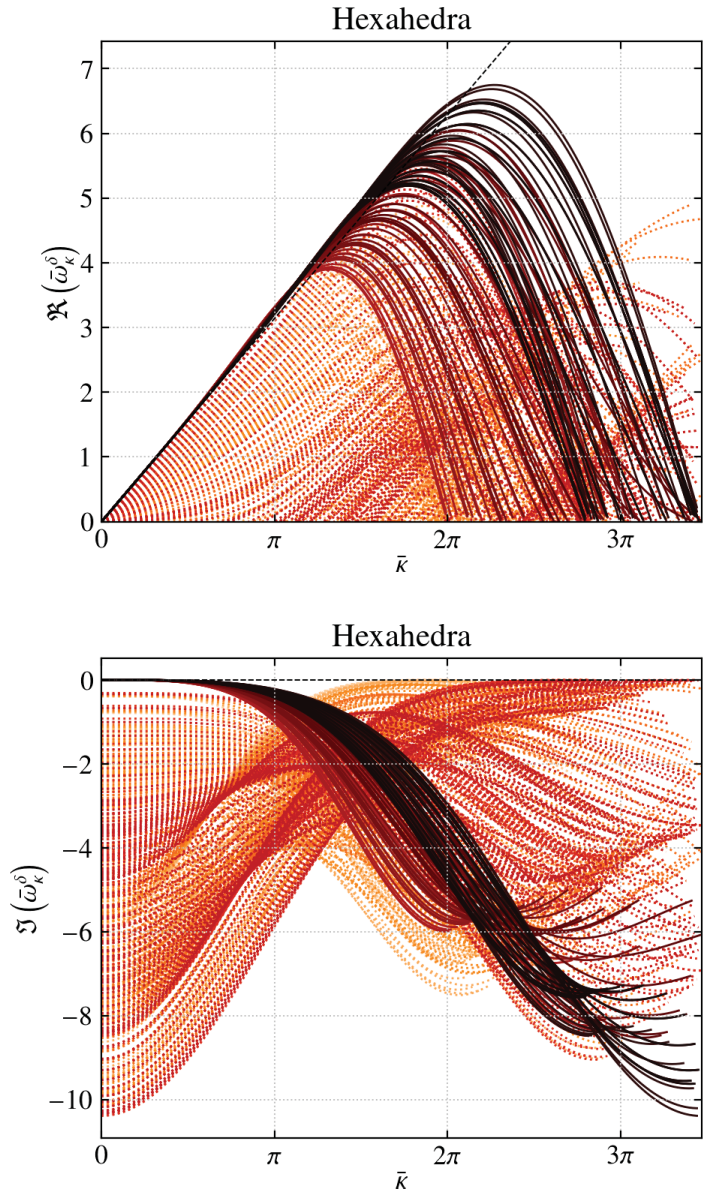


Figure A.1. Concentration of modes for $P1$ spatial discretizations on hexahedra.

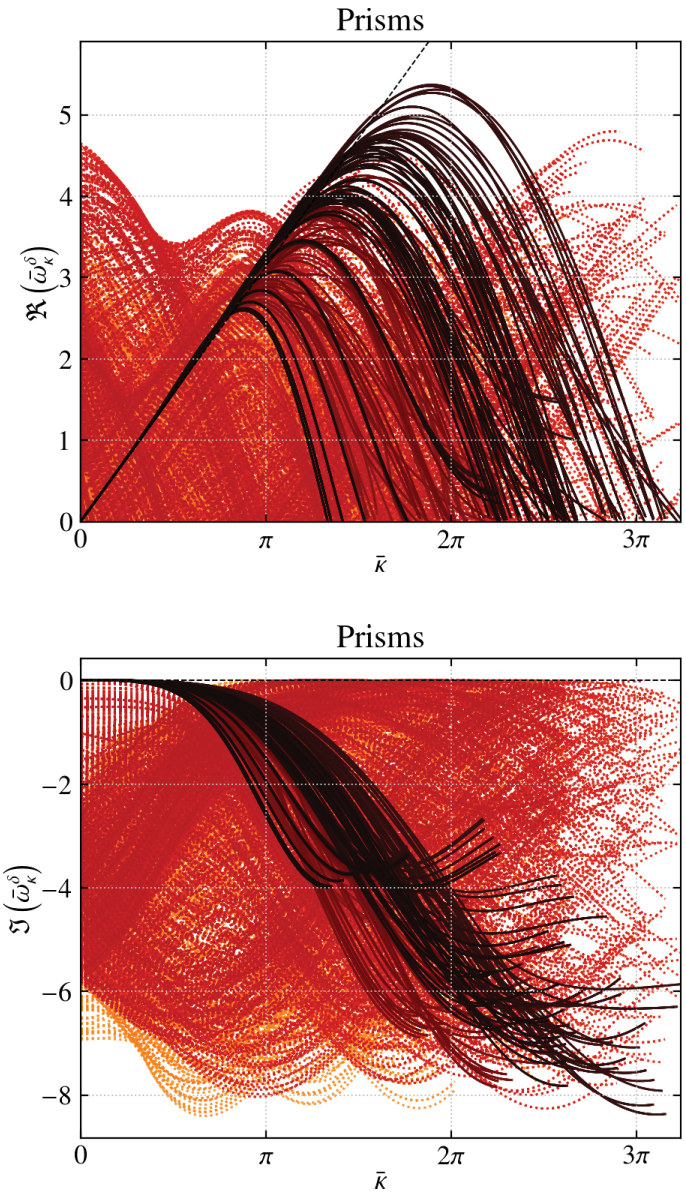


Figure A.2. Concentration of modes for $P1$ spatial discretizations on prisms.

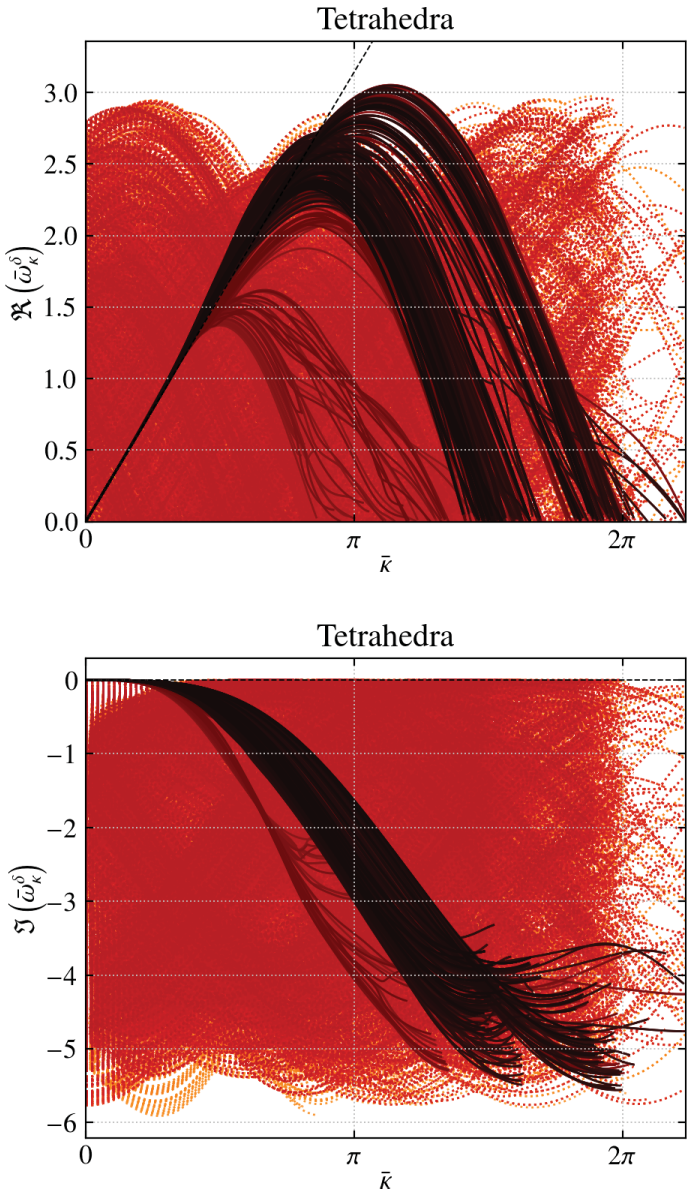


Figure A.3. Concentration of modes for $P1$ spatial discretizations on tetrahedra.

Appendix B

Higher-Order Dispersion Relations

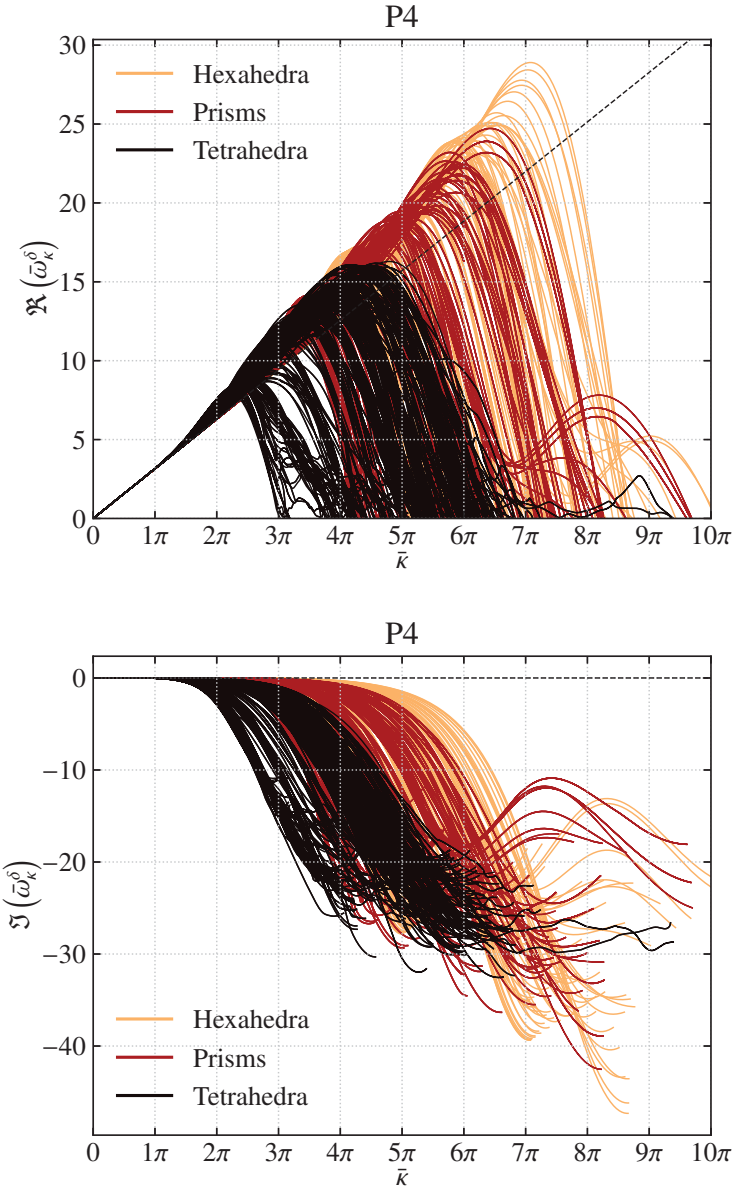


Figure B.1. Numerical dispersion relations for $P4$ schemes using FR.

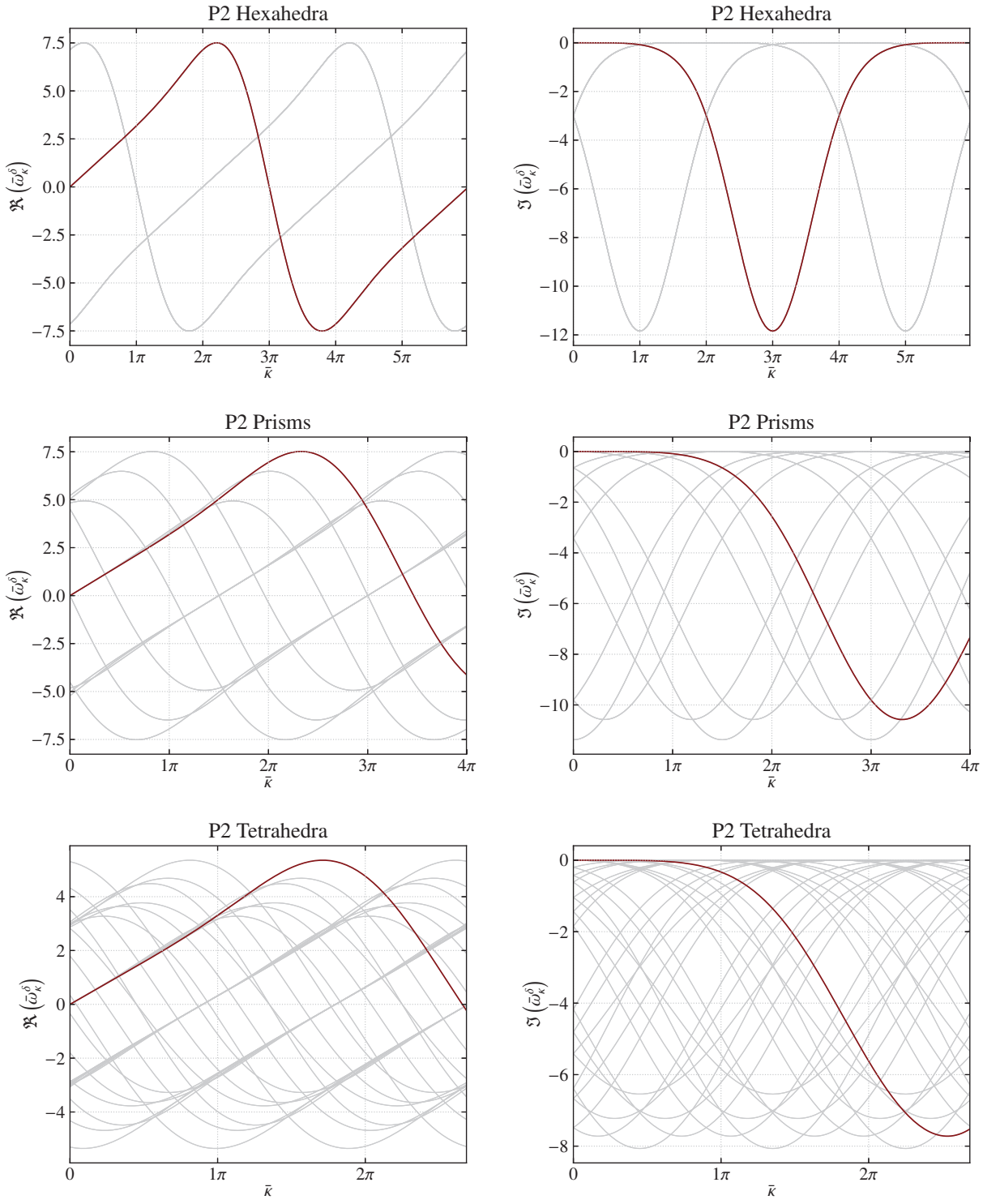


Figure B.2. Eigenmodes for a wave aligned with the x_1 direction on $P2$ schemes.

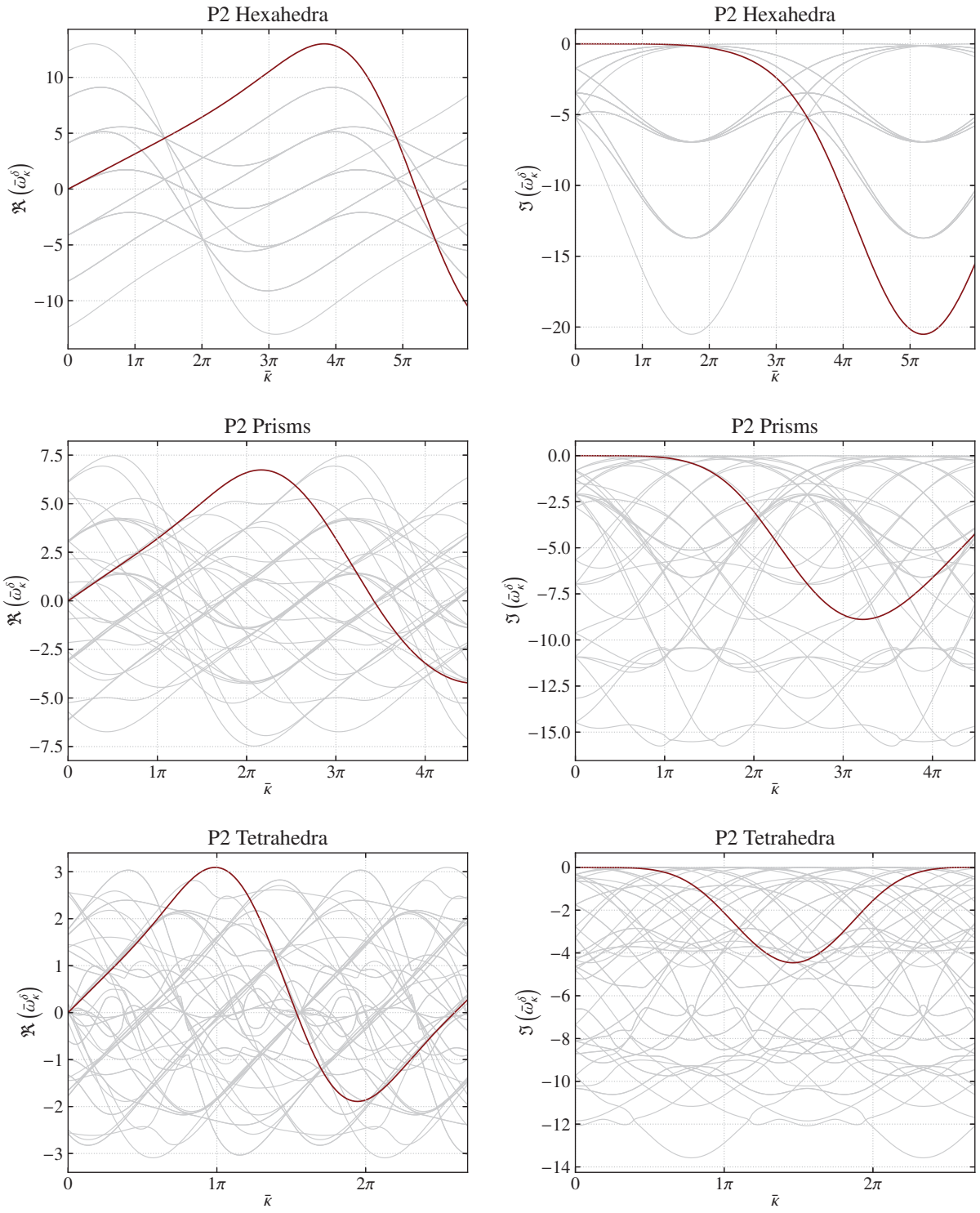


Figure B.3. Eigenmodes for a wave aligned with the main diagonal of the cuboid at $\theta_1 \approx 0.615, \theta_2 = \frac{\pi}{4}$ on $P2$ schemes.

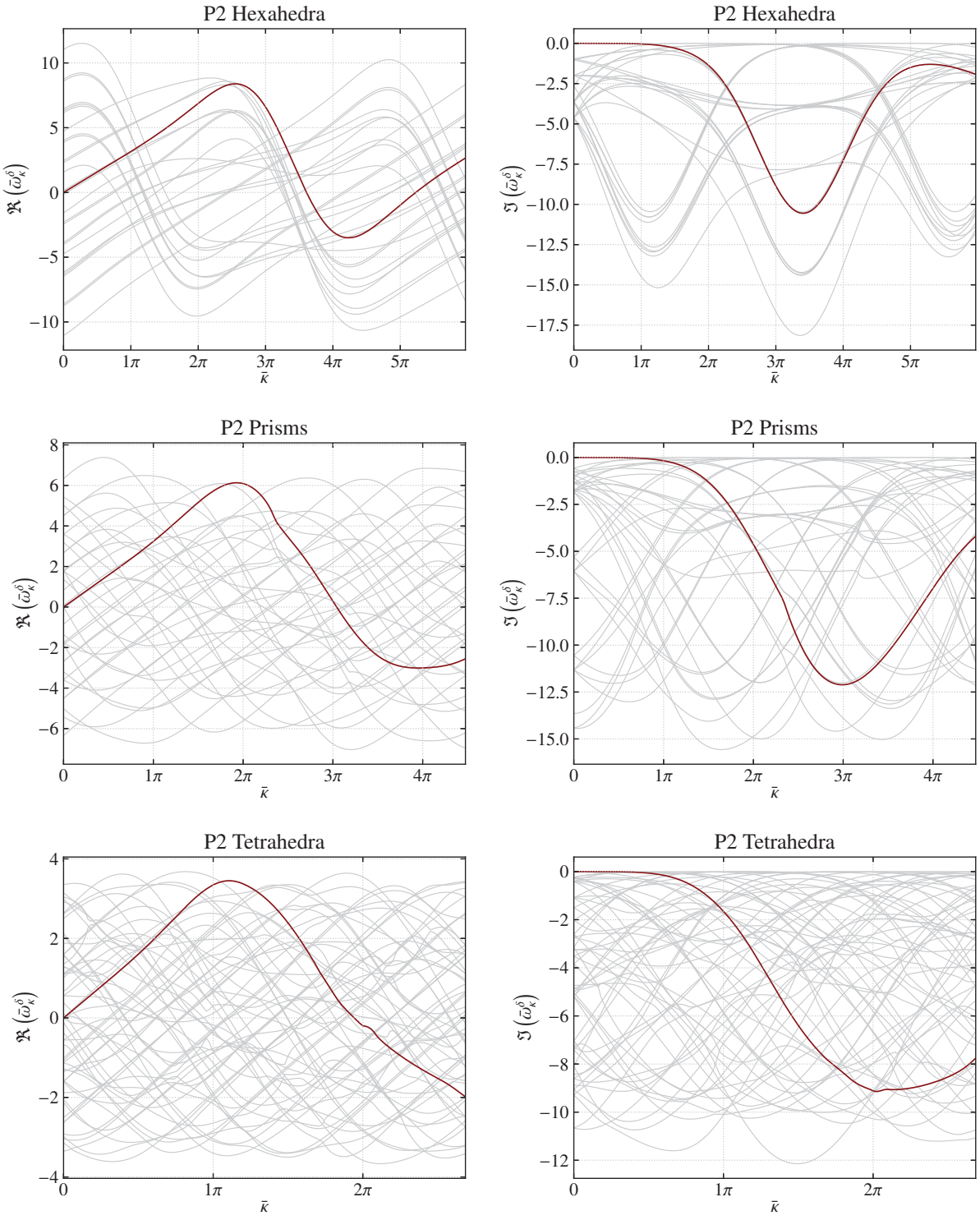


Figure B.4. Eigenmodes for a wave at $\theta_1 = \frac{\pi}{9},\ \theta_2 = \frac{\pi}{9}$ on $P2$ schemes.

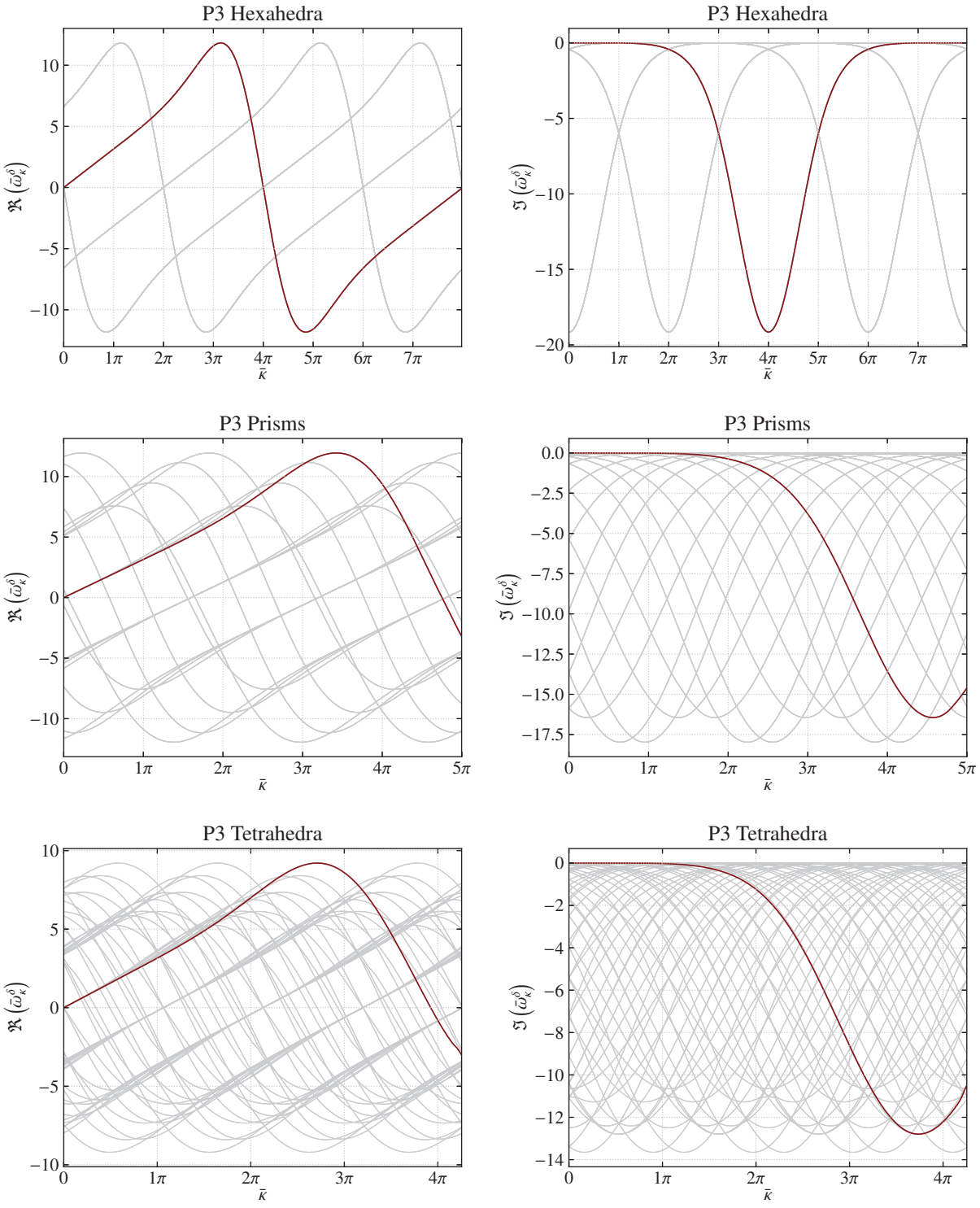


Figure B.5. Eigenmodes for a wave aligned with the x_1 direction on $P3$ schemes.

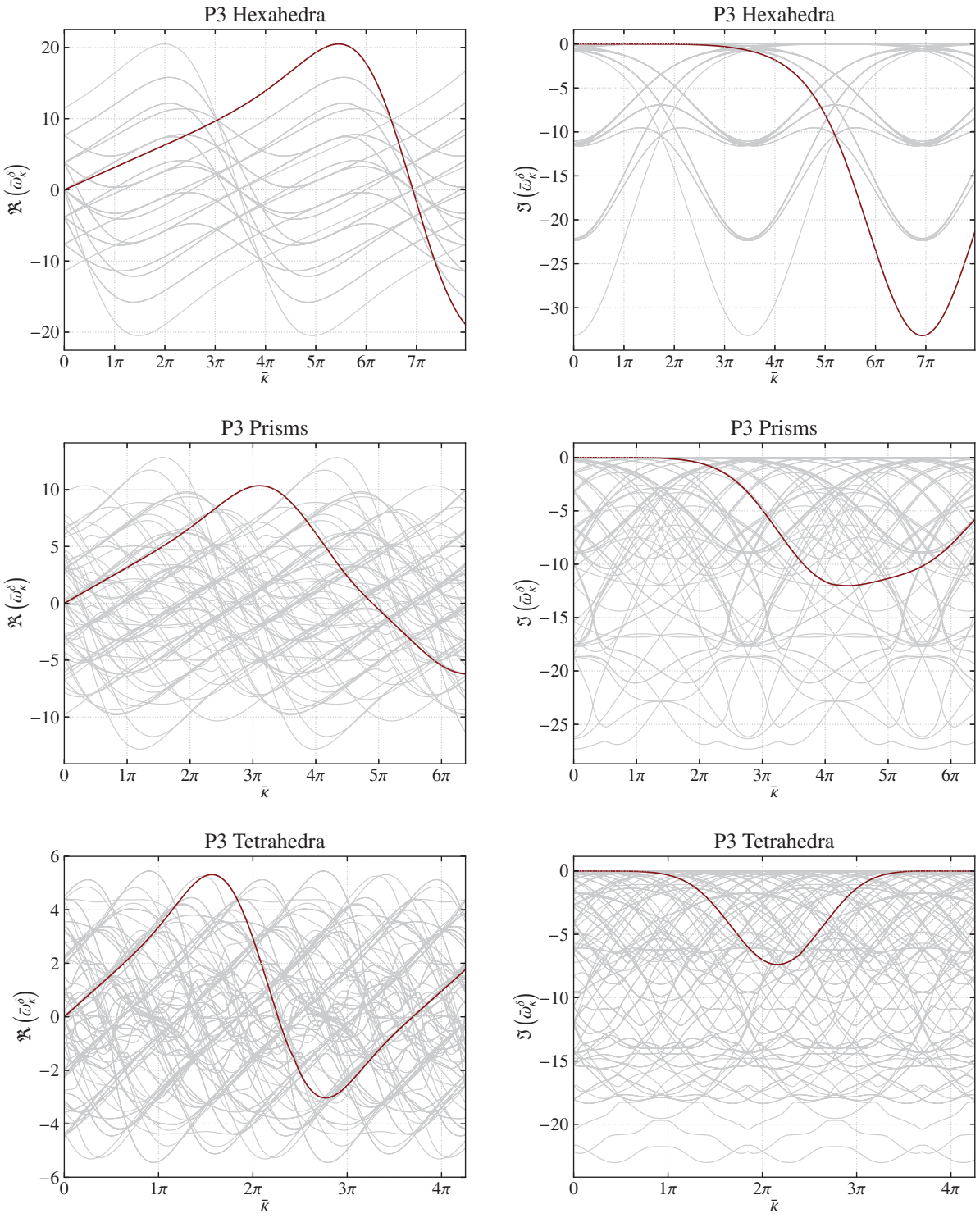


Figure B.6. Eigenmodes for a wave aligned with the main diagonal of the cuboid at $\theta_1 \approx 0.615, \theta_2 = \frac{\pi}{4}$ on $P3$ schemes.

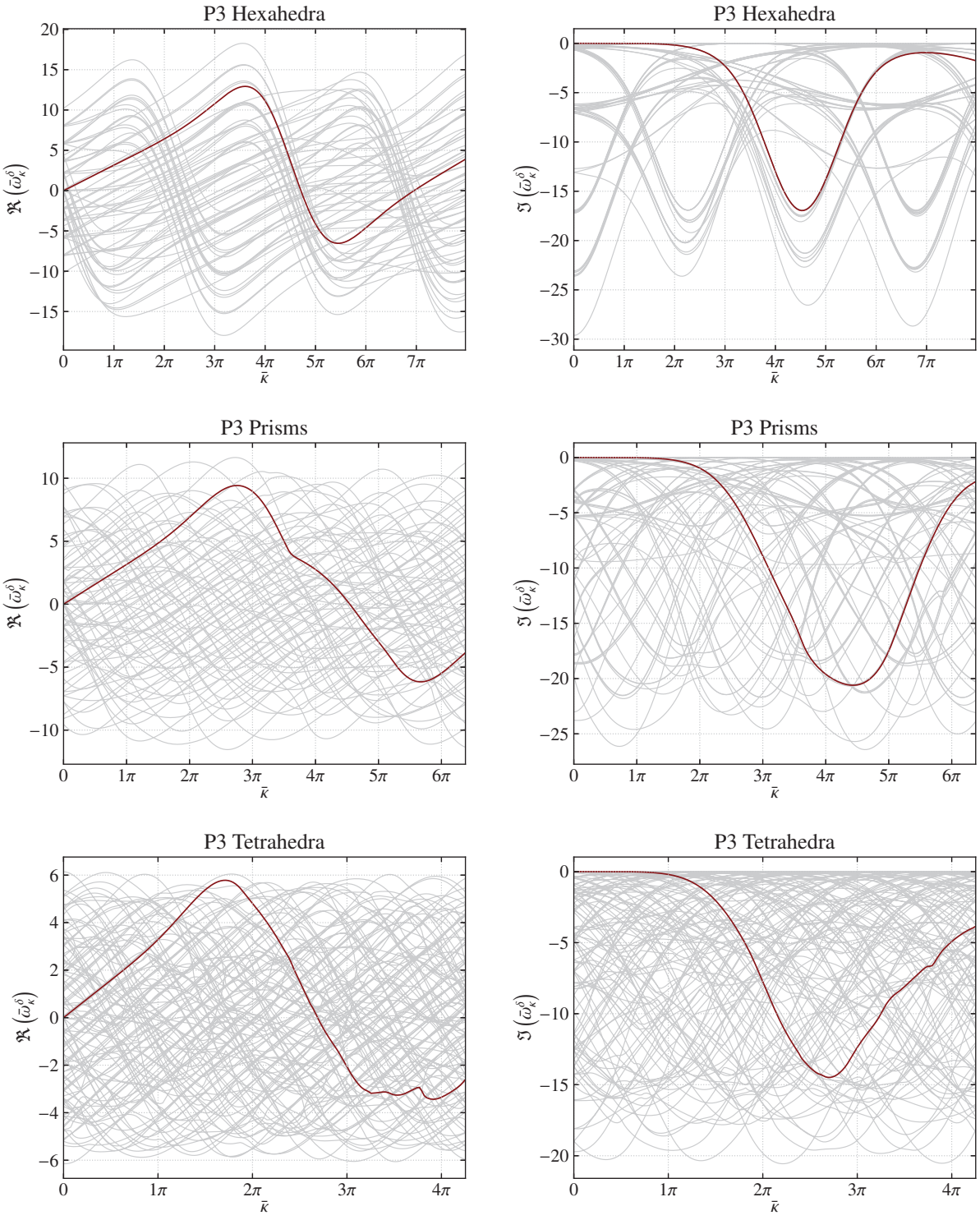


Figure B.7. Eigenmodes for a wave at $\theta_1 = \frac{\pi}{9},\ \theta_2 = \frac{\pi}{9}$ on $P3$ schemes.

Appendix C

Automating CFD Software Verification

Software in constant development requires periodic unit testing to ensure proper functioning of their components. Often programmers set up integration tests to automatically verify the code. Similarly, a CFD solver numerically computes the Navier-Stokes equations and each of its components may be verified to ensure the proper implementation of new algorithms. Given the constant and frequent advancements in the field of computational fluid dynamics, it is not uncommon to periodically improve features related to performance or even accuracy. By doing this, the introduction of human error is likely. In addition, in collaborative environments, even with version control, it is imperative to maintain a set of benchmarks to ensure that all of the components work properly.

In order to achieve this, the development of a simple V&V tool was proposed. Basically, the idea is to run a series of benchmark test cases and ensure that the output files have not had any changes. To quantify this, we make use of the hash of the last written file and compare it with a previously stored one. The V&V tool is written using the Python language, and uses a convenient module called `unittest`. This module does precisely what it name suggests, it provides an API to set up unit testing scripts on the parts of the code we desire.

Listing C.1 shows a snippet of the code. It displays an example unit test using the module from Python with the same name. This module [67] provides a base class `unittest.TestCase` from which we inherit. In addition, the methods `setUp` and `tearDown` are also provided. These two methods are in charge of running any actions before and after the tests have been completed. Since we are dealing with multi-branched, version-controlled software, the initial and final methods are in charge of cloning and compiling the code, and removing the folders after the testing has been performed. The hash of the output files are stored in YAML files [68], which are configure files that are human-readable, easy to modify and provides an interface that integrates very well with Python. These tests are run periodically using a job scheduler. We take advantage of the built-in scheduler

```
class SolverTestCase(unittest.TestCase):
  """Verifies all benchmark cases in CFD solver"""
  def setUp(self):
    # Initialize solver
    self.solver = HORUS()

  def test_equal_hash(self):
    for branch in self.solver.branches:
      # Compile C++ CFD code
      self.solver.compile(branch)

      for case in self.cases:
        # Write CFD output files
        self.solver.run(case)
        ref_hash = get_ref_hash(branch, case)
        output_hash = get_hash()

        # Test hash
        with self.subTests():
          self.assertEqual(ref_hash, output_hash)

  def tearDown(self):
    shutil.rmtree(self.solver.dir)
```

Listing C.1. Unit testing class for verification

in Linux systems, CRON.

Appendix D

Extracting Dispersion Curves

When evaluating the dispersion curves, the extraction of the dominant mode is important for analyzing the numerical error in spatial discretizations. The eigenvalues of the semidiscrete matrix do not follow a specific equation that allows to extract the dispersion modes, which can make the procedure of separating the modes cumbersome. As a result, some authors have generally opted for performing a graphical extraction [15, 21] of the physical eigenvalues. In addition, by means of a modal analysis, a mathematical extraction of the dominant mode was proposed by Vincent et al. [18] for one-dimensional semidiscretizations.

In one dimension, the curves are structured and the procedure can be done both graphically and mathematically. As shown in the spectral analysis in Chapter 5, the numerical dispersion relations for three dimensions may be particularly difficult to obtain mathematically at moderately to poorly resolved wavenumbers. This is a consequence of the dominant eigencurve at low wavenumbers becoming secondary at high wavenumbers. In addition, the non-smooth and dense results for high-order polynomials can make the process of tracing the curves impossible for the human eye. Hence, we propose a simple algorithm to trace the curves using derivatives and some additional considerations.

Algorithm 1 shows the steps that were taken to trace the non-smooth and chaotic modes of three-dimensional semidiscrete systems in this work. Nevertheless, it can be applied to any dimensions, as long as the curves are continuous. The notation and functionality of the methods thereby described are based on existing Numpy functions. The proposed algorithm works for dispersion curves that start at $\kappa = 0$, where the dominant mode must have negligible dissipation. In general, the algorithm starts by a simple sorting of the eigenvalues considering only their imaginary components. With tolerance `dtol`, we define which of these eigenvalues are candidate dominant solutions. Then, we obtain a second set of $\boldsymbol{\omega}^{\delta}$ and append them as well as their derivatives $\left(\boldsymbol{\omega}^{\delta}\right)_{\kappa}$ to the closest value

from the previously sorted modes. From now on, the procedure is a simple tracing of curves by choosing eigenvalues with closest derivatives, since determining by nearest k points may lead to incorrect results as curves tend to cross frequently.

Identifying the dominant mode is performed using a second tolerance value `ctol`. Moura et al [17] found that for wavenumbers in which the amount of dissipation reached 1%, the solution was expected to lose accuracy. Hence, we propose this tolerance value to identify the modes that dissipate first and may be eliminated from the candidate group. And by elimination, as modes become too dissipative, the last mode to remain in the initial array of candidates is considered to be the dominant curve. We remember from the spectral analysis that at certain wave orientations, more than one mode may be concentrated into curves that are identical. Hence, it is necessary to also have this check. We note that the tolerance values can be also somewhat arbitrarily chosen and can lead to failure of the algorithm. In addition, the chosen wavenumbers must be close enough so that modes can be correctly identified.

Algorithm 1: Mode extraction and dominant mode identification for dispersion curves.

```
ω^δ ← Get set of eigenvalues for κ_0 = 0;
W ← Initialize 2D array with eigenvalues sorted by ℑ(ω^δ);
ω^δ ← Get second set of eigenvalues for κ_1 = κ_0 + Δκ;
i ← 0, candidates ← empty array;
for ω_j^* in W do
    if ℑ(ω^δ) < ctol then
        append_to_candidates(i) ;
    ω_j^δ ← nearest_point(ω^δ, ω_j^*);
    (ω^δ)_κ ← Get derivative of ω^δ with nearest eigenvalue;
    append_to_W(ω_j^δ, i);
    append_to_der((ω^δ)_κ, i);
    i+ = 1;
while next ω^δ exists do
    i ← 0;
    for ω_j^* in W[−1, :] do
        (ω^δ)_κ ← Get derivatives of ω_j^* with new set of eigenvalues ω^δ;
        ind ← argmin(abs((ω^δ)_κ));
        append_to_W(ω^δ [ind ], i);
        append_to_der((ω^δ)_κ [ind ], i);
        ω_j^* ← delete_from(ω_j^*, ind);
        i+ = 1;
    if dominant is None then
        secondary ← Get candidate mode with smallest imaginary slope;
        for ω_j^* in W[−1, :] do
            if ℑ(ω_j^*) > dtol then
                append_to_secondary(ω_j^*);
            if len(secondary) == len(candidates) then
                dominant ← Get derivative of candidate eigenvalues;
                if len(dominant)> 1 then
                    check_duplicate(dominant);
                    if no duplicates then
                        raise Exception// No curve was found;
                    else
                        dominant ← dominant [0];
                else
                    candidates ← delete_from(candidates, secondary);
            if dominant is not None then
                W ← re-sort modes so dominant array chooses first;
```